

GUIDA ALLA SOLUZIONE DEI PROBLEMI - OPS 2019

VERSIONE NON DEFINITIVA

INDICE DEL DOCUMENTO

0. INTRODUZIONE	pag. 2
1. ALCUNE REGOLE DI SCRITTURA	pag. 4
2. LA COMPILAZIONE DELLE RISPOSTE	pag. 8
3. PROBLEMI RICORRENTI	pag. 9
4. ELEMENTI DI PSEUDOLINGUAGGIO	pag. 54

0. INTRODUZIONE

Il programma OPS per il 2018-2019 si rivolge, come di consueto, a tre livelli di partecipazione:

- scuola primaria,
- scuola secondaria di primo grado,
- scuola secondaria di secondo grado (primo biennio).

Sono previste *gare a squadre* per tutti i livelli e *gare individuali* per gli ultimi due livelli.

Ogni *gara a squadre* consisterà di norma in una decina di problemi (tale numero potrà variare da 8 a 12 e oltre); la articolazione dei problemi sarà, *usualmente*, la seguente:

1. da quattro a sei problemi formulati in italiano e scelti, di volta in volta, tra l'insieme dei "Problemi ricorrenti" (si veda il successivo elenco);
2. due o tre problemi formulati in italiano e relativi a uno pseudo-linguaggio di programmazione;
3. un problema di comprensione di un testo in lingua italiana;
4. due o tre problemi, in genere formulati in inglese, di argomento ogni volta diverso (almeno in linea di principio).

Ogni *gara individuale* consisterà di norma in una mezza dozzina di problemi (tale numero potrà variare da 5 a 8 e oltre); la articolazione dei problemi sarà, *usualmente*, la seguente:

1. due o tre problemi formulati in italiano e scelti, di volta in volta, tra l'insieme dei "Problemi ricorrenti" (si veda il successivo elenco);
2. due o tre problemi formulati in italiano e relativi a uno pseudo-linguaggio di programmazione;
3. due problemi, in genere formulati in inglese, di argomento ogni volta diverso (almeno in linea di principio).

La difficoltà e la complessità dei problemi saranno commisurate al livello cui tali problemi saranno proposti.

I *Problemi ricorrenti* nelle gare OPS 2018-2019 sono tratti del seguente insieme:

- a) Regole e deduzioni.
- b) Grafi.
- c) *Knapsack*.
- d) Pianificazione.
- e) Statistica elementare.
- f) Relazioni tra elementi di un albero.
- g) Flussi in una rete.
- h) Crittografia.
- i) Movimento di un robot o di un pezzo degli scacchi.
- j) Sottosequenze.
- k) Fatti e conclusioni.

Il seguito di questo documento si articola in quattro parti.

Il paragrafo 1 illustra brevemente come scrivere i numeri, i *termini* e le *liste* (questi ultimi sono "scritture formali" che possono essere utilizzate nei testi dei problemi proposti e nelle soluzioni richieste).

Il paragrafo 2 elenca le regole generali per compilare le risposte ai problemi: tali regole sono *stringenti*, perché la correzione dei problemi viene fatta (normalmente) confrontando la stringa, proposta come risposta ad un quesito, con la stringa che rappresenta la risposta attesa (o "esatta").

Il paragrafo 3 riporta esempi dei problemi ricorrenti; tali esempi sono di norma due: uno orientato alla scuola primaria e uno orientato alla scuola secondaria (se è fornito un solo esempio il problema

ricorrente è orientato a un solo ordine di scuola). I problemi sono di carattere abbastanza semplice; servono principalmente di riferimento per:

- fissare gli argomenti e la terminologia usata,
- tratteggiare i metodi di soluzione,
- mantenere “sintetico” l’enunciato dei problemi assegnati effettivamente in gara.

L’ultimo paragrafo illustra brevemente gli *elementi di pseudolinguaggio* che saranno utilizzati nei problemi assegnati in gara. La trattazione è divisa in due parti: la prima parte riguarda i tre livelli scolastici, la seconda riguarda *prevalentemente* la scuola secondaria (anche se, nel volgere delle gare, potrà interessare anche la scuola primaria). Seguono esempi di problemi riguardanti lo pseudolinguaggio.

N.B. Si suppone che i partecipanti alle gare abbiano letto il presente materiale e, comunque, l’abbiano a disposizione durante lo svolgimento della gara.

1. ALCUNE REGOLE DI SCRITTURA

Nei testi dei problemi possono comparire: *numeri*, *termini* e *liste*; le seguenti osservazioni sono utili per comprenderne il significato e l'uso corretto.

a.0 I NUMERI

I numeri sono scritti sempre, salvo diverso avviso in notazione decimale.

a.1 I NUMERI INTERI

I numeri interi sono stringhe di cifre che, a sinistra, non iniziano con la cifra “0”. Talvolta, *nel testo dei problemi*, se i numeri sono molto lunghi (cioè con almeno cinque cifre), per facilitarne la lettura possono avere un *separatore periodico* ogni tre cifre a partire da destra. Se l'ambito è in *lingua italiana* il separatore è un punto nella parte superiore della riga, ad esempio:

123
1234
12'345
1'234'556'789

Se l'ambito è in *lingua inglese* il separatore è la virgola, ad esempio:

123
1234
12,345
1,234,556,789

N.B. Nella compilazione delle risposte i numeri interi sono *sempre* scritti *senza* separatore periodico e *senza* zeri a sinistra.

a.2 I NUMERI RAZIONALI

I numeri razionali sono scritti con una parte intera e una parte decimale, separate da un marcatore; se l'ambito è in *lingua italiana* il marcatore è la *virgola* nella parte bassa del rigo:

0,23
2,343
45,0

se l'ambito è in *lingua inglese* il marcatore è il *punto* nella parte bassa del rigo:

0.23
2.343
45.0

I numeri razionali non devono iniziare (a sinistra) con uno zero.

A destra lo zero è ammesso se è l'unica cifra a destra della virgola, oppure quando il testo del problema specifica il numero di decimali dopo la virgola. Per esempio $9/5$ è uguale a 1,8; se però il problema richiede esplicitamente due decimali allora si scriverà 1,80.

N.B. Nella compilazione delle risposte il numero *intero* 45 è “diverso” dal numero *decimale* 45,0 (o 45.0 se l'ambito è in lingua inglese): il testo del problema specificherà sempre la natura della risposta richiesta.

a.3 TRONCAMENTO E ARROTONDAMENTO DEI NUMERI RAZIONALI

Le due nozioni sono bene illustrate dai seguenti esempi.

Esempio 1. Supponiamo che, nella soluzione di un problema, il risultato (ottenuto con opportuni calcoli) sia 9,3472; se nella risposta sono richieste, *due cifre decimali arrotondate*: occorre scrivere 9,35; se invece sono richieste *due cifre decimali troncate*, occorre scrivere 9,34.

Es. 2. Supponiamo che, nella soluzione di un problema, il risultato (ottenuto con opportuni calcoli) sia 9,3442; se nella risposta sono richieste, *due cifre decimali arrotondate*: occorre scrivere 9,34; se invece sono richieste *due cifre decimali troncate*, occorre scrivere 9,34.

Altri esempi, sempre riferiti a due cifre decimali, sono raccolti nella seguente tabella.

Numero calcolato	Numero arrotondato	Numero troncato
9,3435	9,34	9,34
9,3402	9,34	9,34
8,9951	9,00	8,99
0,898	0,90	0,89

a.4 DATE ED ORE

Il testo del problema e/o le diciture che accompagnano i campi da riempire, specificheranno in maniera puntuale come indicare date e ore.

b.0 I TERMINI

Un *termine* è una scrittura del tipo:

minerale(m1,90,300)

attività(alfa,4,6)

cioè un *nome* seguito da *uno* o più *argomenti* racchiusi tra parentesi e separati da virgole (se sono più di uno).

N.B. Nella scrittura di un termine non intervengono spazi.

Il *nome* è una stringa di caratteri che inizia con una lettera; gli *argomenti* sono parole, sigle o numeri interi.

b.1 I TERMINI: DEFINIZIONE ED ESEMPI D'USO

In generale, prima dell'uso, ogni termine è definito. La definizione indica il *nome* del termine, gli *argomenti* e il loro *significato*.

Si consideri, ad esempio, la tabella seguente che si riferisce ad un magazzino di minerali: riporta la sigla il peso e il valore di ogni esemplare presente nel magazzino.

minerale		
sigla	peso in chili	valore in euro
m1	30	213
m2	35	200
m3	38	230

Il contenuto della tabella può essere descritto con dei termini; ogni termine è così *definito*:

minerale(<sigla>,<peso in chili>,<valore in euro>)

La *definizione* mostra il *nome* del termine, il *numero* degli *argomenti* e il loro *significato*. Nel caso delle tabelle il nome del termine è il nome della tabella, il numero degli argomenti è il numero delle colonne e il significato di ogni termine è l'intestazione della colonna. Il contenuto della tabella può essere così descritto:

minerale(m1,30,213)

minerale(m2,35,200)

minerale(m3,38,230)

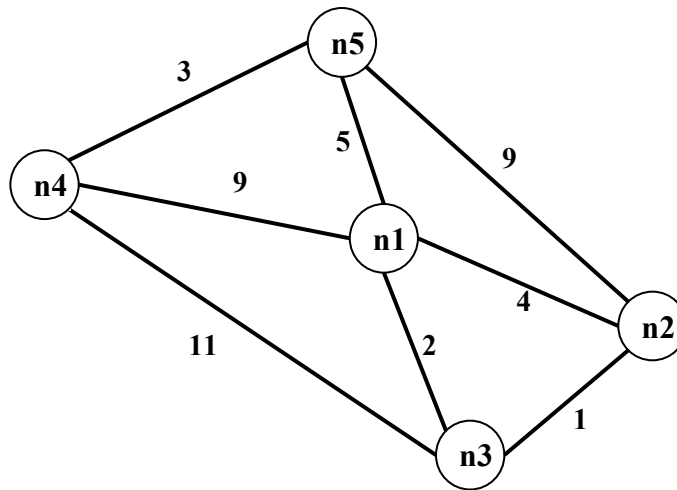
Come ulteriore esempio si consideri la seguente figura che rappresenta un *grafo*; questo è composto da *nodi*, distinti da una sigla, e da *archi* che congiungono i nodi.

Un grafo si può pensare come la rappresentazione schematica di città congiunte da strade.

Un grafo può essere descritto con dei termini; ogni termine è associato a un arco ed è così *definito*:

arco(<nodo1>,<nodo2>,<lunghezza>)

La definizione mostra il nome del termine e il numero degli argomenti (tre): i primi due si riferiscono ai nodi posti agli estremi dell'arco, il terzo descrive la lunghezza dell'arco.



I termini che descrivono il grafo in figura sono:

arco(n1,n2,4)

arco(n1,n3,2)

arco(n1,n4,9)

arco(n1,n5,5)

arco(n4,n5,3)

arco(n4,n3,11)

arco(n2,n3,1)

arco(n2,n5,9)

c.0 LE LISTE

Una *lista* è una scrittura del tipo:

[alfa,beta,gamma]

[]

[a1,2,56,b2,3,67]

cioè una coppia di parentesi quadre che racchiudono un certo numero di *elementi* separati da virgole (se sono più di uno). Una lista può non contenere elementi: in tal caso si dice *lista vuota*.

N.B. Nella scrittura di una lista non intervengono spazi.

Gli *elementi* sono parole, sigle, numeri interi o *altre liste*.

c.1 LE LISTE: ESEMPI D'USO

Si consideri la seguente figura che rappresenta una scacchiera 8×8. Facendo riferimento allo spigolo in basso a sinistra, ogni casella può essere individuata da una coppia di coordinate, scritte come una lista, il cui primo elemento è la "X", cioè lo spostamento orizzontale e il secondo elemento è la "Y", cioè lo spostamento verticale.

Per esempio il numero 5 è nella casella [3,2]; la bandiera è nella casella [1,8] e il robot nella casella [8,1].

♁			■				
	■	■			11		
						■	
					12		
		5		■		13	
		■					♁

Per indicare la posizione sulla scacchiera dei quadrati neri si può usare una lista i cui elementi sono le coordinate delle caselle che li contengono; si ottiene, quindi, una lista di liste:

`[[2,5],[3,1],[3,5],[4,8],[5,2],[7,4]]`

Volendo indicare la posizione sulla scacchiera e il valore dei quattro numeri si può usare una lista di quattro elementi che, a loro volta, sono liste; ciascuna lista “interna” è formata dalle coordinate della casella che contiene il numero seguite dal valore:

`[[3,2,5],[6,3,12],[6,5,11],[7,2,13]]`

Per indicare la posizione della freccia → sulla scacchiera si può usare la lista vuota:

`[]`

per significare, appunto, che non compare.

2. LA COMPILAZIONE DELLE RISPOSTE

Si tenga presente che le risposte ai problemi vengono valutate da un sistema automatico che, essenzialmente, confronta stringhe: quindi tali risposte vanno compilate rispettando *esattamente* le indicazioni riportate nel testo dei problemi; inoltre, si devono sempre osservare le seguenti regole generali.

1. In ciascun campo disponibile per la risposta è possibile, *di norma*, digitare, allineato a sinistra, un solo elemento: parola, numero, lista o termine; le parole devono essere scritte *senza accenti*; per i numeri si veda il paragrafo precedente.
2. Talvolta il testo del problema può specificare che la risposta è una frase: in tal caso occorre scriverla con *un solo spazio* tra due parole consecutive.
3. Gli elementi che compongono una lista vanno riportati fra parentesi quadre separati da virgole *senza spazi*: per esempio la lista delle prime tre lettere (minuscole) dell'alfabeto va scritta nel modo seguente: [a,b,c] e non [a, b, c]; i campi, per le risposte che richiedono liste, riportano già le parentesi quadre “esterne”; se gli elementi delle lista sono, a loro volta, liste occorre scrivere le opportune parentesi che li racchiudono.
4. L'elevazione a potenza è indicata da x^n ; per esempio: x^2 deve essere scritto x^2 .
5. Una frazione deve essere scritta su una linea; per esempio: $2/3$.
6. Se nel testo del problema è indicata una particolare modalità di risposta, per esempio:
 digitare V per vero o F per falso
 è necessario attenersi strettamente alle istruzioni: in caso contrario il sistema registrerà una risposta errata.
7. Gli allenamenti disponibili sul sito consentono di familiarizzarsi con le regole di compilazione delle risposte.

3. PROBLEMI RICORRENTI

a) REGOLE E DEDUZIONI

PREMESSA

Per risolvere problemi spesso esistono delle regole che, dai dati del problema, permettono di calcolare o *dedurre* la soluzione.

Questa situazione si può descrivere col termine

regola(<sigla>,<lista degli antecedenti>,<conseguente>)

che indica una regola di nome <sigla> che consente di dedurre <conseguente> conoscendo tutti gli elementi contenuti nella <lista degli antecedenti>, detta anche *premessa*. Problemi “facili” possono essere risolti con una sola regola; per problemi “difficili” una sola regola non basta a risolverli, ma occorre applicarne diverse in successione.

Si considerino le seguenti regole (a rigore le regole associate ai seguenti termini):

regola(1,[e,f],b) regola(2,[m,f],e) regola(3,[m],f)
regola(4,[b,f],g) regola(5,[b,g],c) regola(6,[g,f],c)

Per esempio la regola 1 dice che si può calcolare (o dedurre) **b** conoscendo **e** ed **f** (cioè gli elementi della lista [e,f]); conoscendo **b** ed **f** (cioè gli elementi della lista [b,f]) è possibile dedurre **g** con la regola 4. Quindi, a partire da **e** ed **f** è possibile dedurre prima **b** (con la regola 1) e poi **g** (con la regola 4).

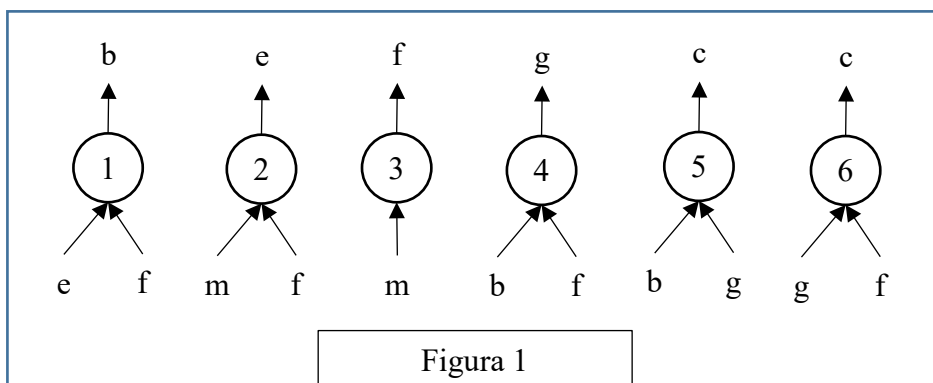
N.B. I due seguenti termini:

regola(1,[e,f],b)
regola(1,[f,e],b)

individuano la stessa regola, che permette di dedurre **b** da **e** ed **f** (o da **f** e da **e**).

Un *procedimento di deduzione* (o deduttivo, o di calcolo) è rappresentato da un *insieme di regole da applicare in sequenza opportuna* per dedurre un certo elemento (incognito) a partire da certi dati: quindi può essere descritto dalla lista delle sigle di queste regole. Il procedimento [1,4] descrive la soluzione del problema: “dedurre **g** a partire da **e** ed **f**”.

Una maniera grafica per rappresentare le regole è quella mostrata nella seguente figura 1: consiste nell’associare un albero (rovesciato) ad ogni regola: la radice (in alto) è il conseguente, le foglie (in basso) sono gli antecedenti.



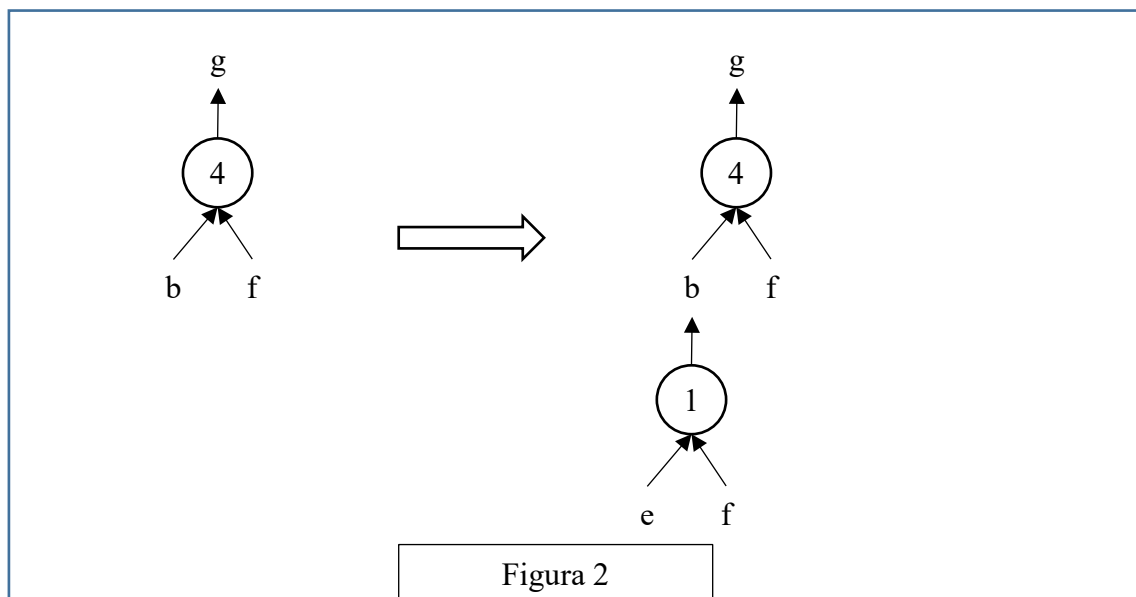
Con questa rappresentazione grafica, risolvere il problema “dedurre **g** a partire da **e** ed **f**” è particolarmente facile; si cerca un “albero” (cioè una regola) che ha come radice l’incognita (cioè **g**): in questo caso ne esiste solo uno che è la regola 4: si veda la figura 2 a sinistra.

Le foglie di questo albero (**b** ed **f**) *non* sono tutte note: quelle note (**f** in questo caso) sono vere e proprie foglie, quelle incognite (**b** in questo caso) vanno considerati come “anelli” a cui “appende-

re” un altro albero; quindi bisogna continuare *sviluppando* la foglia incognita **b**, cioè “*appendendo*” a **b** l’albero rappresentato dalla regola 1, come illustrato nella figura 2 a destra.

Adesso tutte le foglie dell’albero così ottenuto (**e** ed **f**) sono note e il problema è risolto.

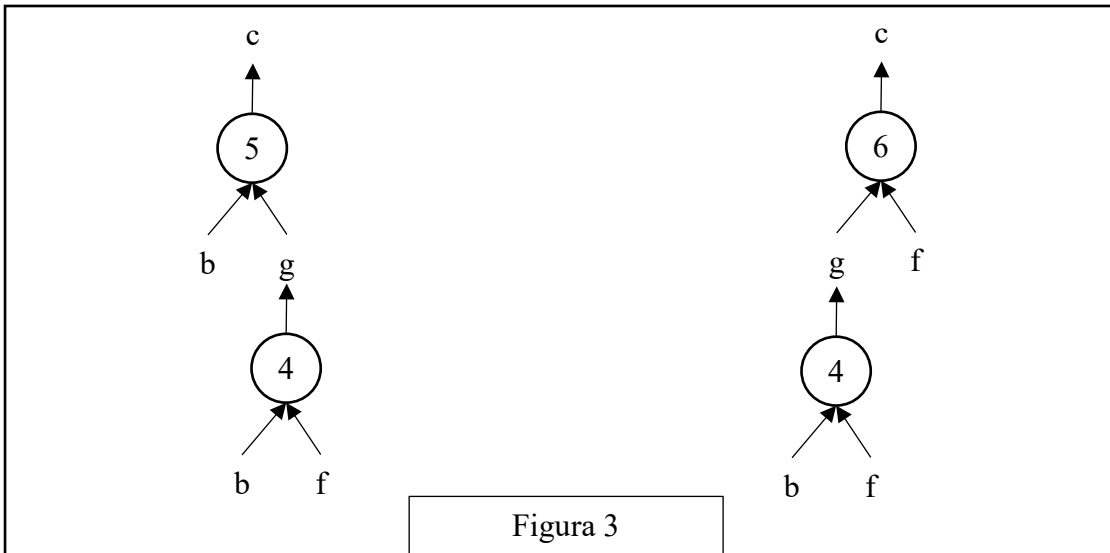
Si può anche dire che un albero le cui foglie sono tutte note rappresenta un procedimento per dedurre la “radice” a partire dalle “foglie”. Per costruire la lista corrispondente occorre *partire dal basso*: prima si applica la regola 1, che utilizza solo i dati; poi si può applicare la regola 4. Il procedimento è quindi (individuato dalla lista) [1,4].



N.B. Nelle liste richieste occorre elencare le sigle delle regole nell’ordine che corrisponde alla sequenza di applicazione: la prima (a sinistra) della lista deve essere la sigla che corrisponde alla prima regola da applicare (che ha come antecedenti solo dati); l’ultima (a destra) deve essere la sigla della regola che ha come conseguente l’elemento incognito da dedurre.

Nella lista non ci sono regole *ripetute* (infatti un procedimento di deduzione è un *insieme* di regole da applicare in opportuna sequenza). L’applicazione di una regola rende disponibile il conseguente da utilizzare (come antecedente) nell’applicazione di regole successive.

La lista associata a un (ben preciso) procedimento si costruisce quindi per passi successivi a partire dal primo elemento che è la sigla della prima regola da applicare; ad ogni passo, se ci fossero più regole applicabili, occorre dare la precedenza (nella lista) a quella con sigla *inferiore* (questo per rendere *unica* la lista associata al procedimento).



N.B. In alcuni casi esistono più procedimenti deduttivi possibili che permettono di ricavare un certo elemento dagli stessi dati, in maniere diverse (cioè con alberi diversi e quindi con insiemi diversi di regole). Per esempio il problema “dedurre **c** a partire da **b** ed **f**” (dalle regole viste sopra) ha due distinti procedimenti risolutivi; gli alberi relativi ai due procedimenti sono mostrati nella seguente figura 3.

Le liste associate sono, rispettivamente, [4,5] e [4,6].

In un procedimento deduttivo, il numero di regole *differenti* coinvolte (e, quindi, anche il numero di elementi della lista corrispondente al procedimento) si dice *lunghezza* del procedimento.

PROBLEMA 1

Siano date le seguenti regole:

regola(1,[b,c],a) regola(2,[c,d],a)
 regola(3,[b,c,d],a) regola(4,[b,a],f)

Trovare:

1. la sigla N della regola che consente di dedurre **a** da **d** e **c**;
2. la lista L che rappresenta il procedimento per dedurre **f** da **b** e **c**.

Scrivere le soluzioni nella seguente tabella.

N	
L	

SOLUZIONE

N	2
L	[1,4]

COMMENTI ALLA SOLUZIONE

Per rispondere alle due semplici domande è opportuno applicare il metodo *backward*, cioè è opportuno partire dalla incognita (l’elemento che occorre dedurre) e cercarlo nel conseguente delle varie regole.

Per la prima domanda (che chiede di dedurre **a**) si osservi che le regole 1, 2 e 3 hanno tutte come conseguente **a**, e quindi permettono di dedurlo; ma solo la regola 2 ha come premessa [c,d] cioè ha come antecedenti i dati (**d** e **c**) della prima domanda: quindi è quella la regola cercata.

Per rispondere alla seconda domanda, si osservi come una sola regola, la 4, ha come conseguente **f**; ma ha come antecedenti **b** e **a**; però, di questi, solo **b** è noto: quindi, per poter applicare tale regola, occorre dedurre **a**; come già osservato, le regole 1, 2 e 3 hanno tutte come conseguente **a**, e quindi permettono di dedurlo; ma adesso (a differenza della prima domanda) sono noti **b** e **c**: quindi stavolta è la regola 1 quella che deve essere applicata. Nel costruire la lista richiesta si ricordi che il primo elemento di tale lista è la prima regola che deve essere applicata, quindi (tutti) i suoi antecedenti devono essere dati.

PROBLEMA 2

Siano date le seguenti regole:

regola(1,[u,d],c)	regola(2,[q,n],g)	regola(3,[p,q],n)
regola(4,[c,d],z)	regola(5,[u],d)	regola(6,[n,u],m)

Trovare:

1. la lista L1 che descrive il procedimento per dedurre **g** a partire da **p** e **q**;
2. la lista L2 che descrive il procedimento per dedurre **z** a partire da **u**.

L1	
L2	

SOLUZIONE

L1	[3,2]
L2	[5,1,4]

COMMENTI ALLA SOLUZIONE

Per risolvere questo tipo di problemi si può usare il metodo *backward* (o *top down*) che consiste nel partire dalla incognita e cercare di individuare una regola per derivarla. Se esiste una regola i cui antecedenti sono tutti noti (i dati) la soluzione è trovata; altrimenti si cerca una regola i cui antecedenti non sono tutti noti e si continua a cercare regole per derivare gli antecedenti incogniti (che compaiono nella premessa).

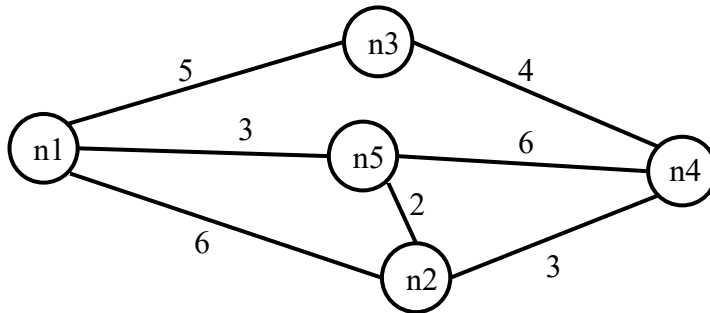
Per la prima domanda si verifica immediatamente che **g** compare come conseguente nella regola 2 che ha come antecedenti **q** (dato) e **n** (incognito). Occorre quindi dedurre **n**: questo è conseguente solo della regola 3, che ha come antecedenti **p** e **q** entrambi dati. Quindi la lista L1 è [3,2]; si noti l'ordine delle regole: la prima che compare (a sinistra) nella lista è la prima da applicare e l'ultima trovata col metodo *backward*.

Per la seconda domanda, di nuovo si può osservare che **z** compare come conseguente nella regola 4 che ha come antecedenti **c** e **d**: entrambi incogniti. È facile vedere che **d** può essere dedotto con la regola 5 da **u** (noto); poi noto anche **d**, con la regola 1 si deduce **c**. Quindi la lista L2 è [5,1,4].

N.B. La prima regola che compare nella lista (che rappresenta il procedimento) ha come antecedenti solo dati; la seconda e le successive hanno antecedenti presi dai dati o dagli elementi dedotti mediante le regole che compaiono precedentemente nella lista. L'ultima regola ha come conseguente l'elemento cercato.

b) GRAFI**PREMESSA**

Un *grafo* si può pensare come l'astrazione di una carta geografica: per esempio il seguente grafo descrive i collegamenti esistenti fra alcune (5) città: queste sono rappresentate da *nodi* di nome n_1, n_2, \dots, n_5 e i collegamenti sono rappresentati da segmenti tra i nodi, detti *archi*. A ogni arco è associata una lunghezza, come illustrato in figura.



Questo grafo può essere descritto da un elenco di termini, ciascuno dei quali definisce un arco tra due nodi del grafo con la indicazione della relativa distanza in chilometri:

arco($n_1, n_2, 6$) arco($n_1, n_3, 5$) arco($n_3, n_4, 4$)
 arco($n_1, n_5, 3$) arco($n_2, n_4, 3$) arco($n_2, n_5, 2$)
 arco($n_5, n_4, 6$)

Due nodi si dicono *adiacenti* se sono collegati da un arco.

Il numero di archi che “escono” da un nodo si dice *valenza* del nodo; per esempio nel grafo in figura, il nodo n_3 ha valenza 2, gli altri hanno valenza 3.

Un *percorso* (o *cammino*) tra due nodi del grafo consiste in una sequenza di nodi ciascuno dei quali (tranne l'ultimo) è adiacente con il successivo; un percorso può, quindi essere descritto con una lista di nodi (quelli toccati dal percorso, ordinata dal nodo di partenza al nodo di arrivo). Per esempio, la lista $[n_5, n_2, n_4, n_3]$ descrive un percorso dal nodo n_5 al nodo n_3 ; tale percorso ha lunghezza $K = 2 + 3 + 4 = 9$.

Un *ciclo* è un percorso che inizia e termina nello stesso nodo, per esempio $[n_5, n_2, n_1, n_5]$.

Un percorso si dice *semplice* se *non* ha nodi ripetuti: un percorso semplice, quindi, non contiene cicli; per esempio $[n_5, n_2, n_4, n_3]$ è semplice, mentre $[n_5, n_2, n_1, n_5, n_2, n_4, n_3]$ non è semplice perché ha nodi ripetuti.

N.B. Dato un grafo, come quello della precedente figura, è facile scrivere l'insieme di termini che lo descrivono; viceversa, disegnare il grafo da un insieme dei termini è meno ovvio (si vedano i problemi seguenti).

In questo contesto vengono proposti due tipi di problemi: quelli che trattano di percorsi e quelli che trattano di adiacenze.

PROBLEMA 1

È dato un grafo descritto dal seguente elenco di archi:

arco($n_1, n_2, 2$) arco($n_2, n_3, 2$) arco($n_3, n_1, 5$)
 arco($n_4, n_1, 1$) arco($n_4, n_2, 4$) arco($n_4, n_5, 3$)

Disegnare il grafo e trovare:

- la lista L_1 del percorso più breve tra n_5 e n_3 e calcolarne la lunghezza K_1 ;
- la lista L_2 del percorso più lungo (senza passare più volte per uno stesso nodo) tra n_5 e n_3 e calcolarne la lunghezza K_2 .

L1	
K1	
L2	

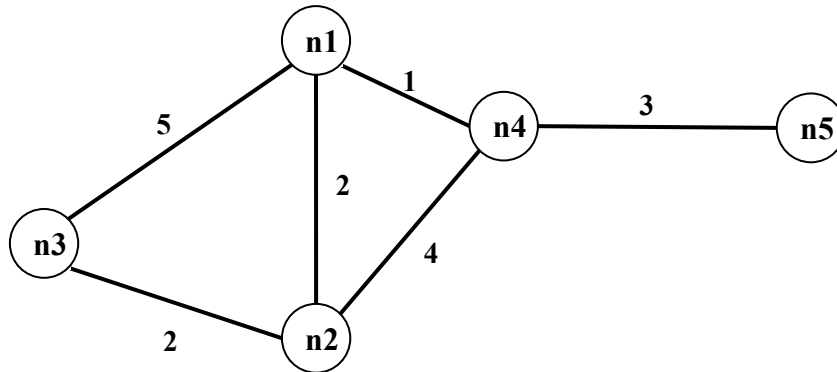
K2	
----	--

SOLUZIONE

L1	[n5,n4,n1,n2,n3]
K1	8
L2	[n5,n4,n2,n1,n3]
K2	14

COMMENTI ALLA SOLUZIONE

Per disegnare il grafo si osservi innanzitutto che vengono menzionati 5 nodi (n1, n2, n3, n4, n5); si procede per tentativi: si disegnano i 5 punti nel piano e li si collega con archi costituiti da segmenti: probabilmente al primo tentativo gli archi si incrociano; si cerca poi di risistemare i punti in modo da evitare gli incroci degli archi: spesso questo si può fare in più modi. Da ultimo si riportano le distanze sugli archi, come mostrato dalla figura seguente.



Si noti che le lunghezze degli archi che compaiono nei termini (che rappresentano delle strade) *non* sono (necessariamente) proporzionali a quelle degli archi del grafo (che sono, segmenti di retta). Per rispondere alle due domande occorre elencare sistematicamente *tutti* i percorsi, che non passino più volte per uno stesso punto, tra n5 e n3:

PERCORSO da n5 a n3	LUNGHEZZA
[n5, n4, n1, n2, n3]	3+1+2+2=8
[n5, n4, n1, n3]	3+1+5=9
[n5, n4, n2, n3]	3+4+2=9
[n5, n4, n2, n1, n3]	3+4+2+5=14

L1, K1, L2, K2 seguono immediatamente.

PROBLEMA 2

Un grafo (che corrisponde alla rete di strade che collegano delle città) è descritto dal seguente elenco di archi:

a(n1,n2,13)	a(n2,n3,3)	a(n3,n4,13)	a(n1,n4,3)
a(n4,n5,3)	a(n5,n1,5)	a(n2,n5,7)	a(n3,n5,11)

Disegnare il grafo e trovare:

1. la lista L1 del percorso semplice più breve tra n1 e n3;

2. la lista L2 del percorso semplice più lungo tra n1 e n3.

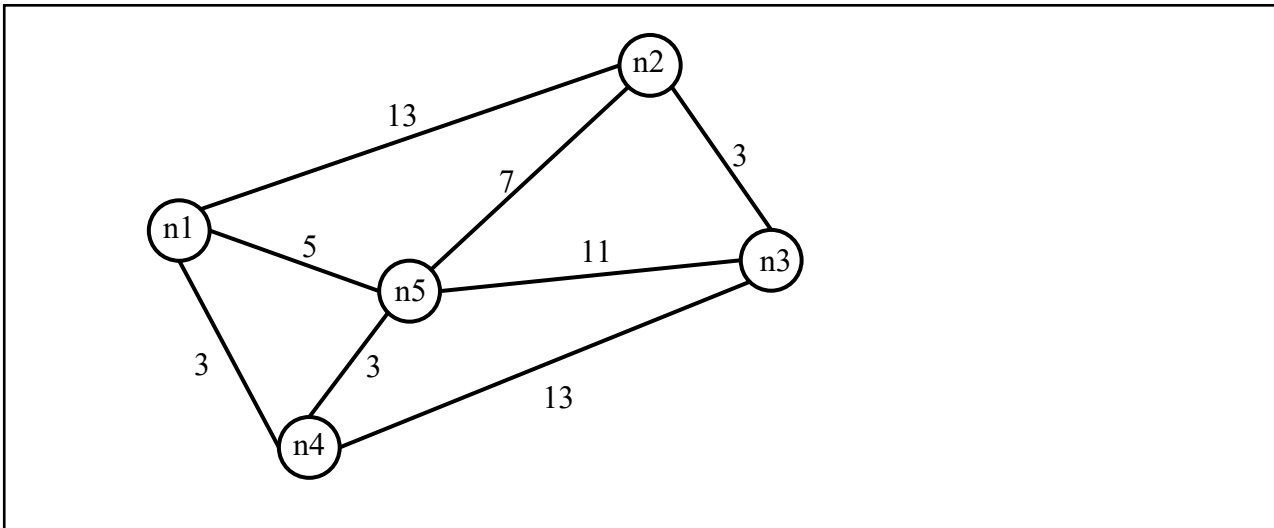
L1	
L2	

SOLUZIONE

L1	[n1, n5, n2,n3]
L2	[n1, n2, n5, n4, n3]

COMMENTI ALLA SOLUZIONE

Per disegnare il grafo si osservi innanzitutto che vengono menzionati 5 nodi (n1, n2, n3, n4, n5); si procede per tentativi: si disegnano i 5 punti nel piano e li si collega con archi rettilinei: probabilmente al primo tentativo gli archi si incrociano; si cerca poi di risistemare i punti in modo da evitare gli incroci degli archi: spesso questo si può fare in più modi. Da ultimo si riportano le distanze sugli archi, come mostrato dalla figura seguente.



Si noti che le lunghezze degli archi che compaiono nei termini (che rappresentano delle strade) *non* sono necessariamente proporzionali a quelle degli archi del grafo (che sono segmenti di retta).

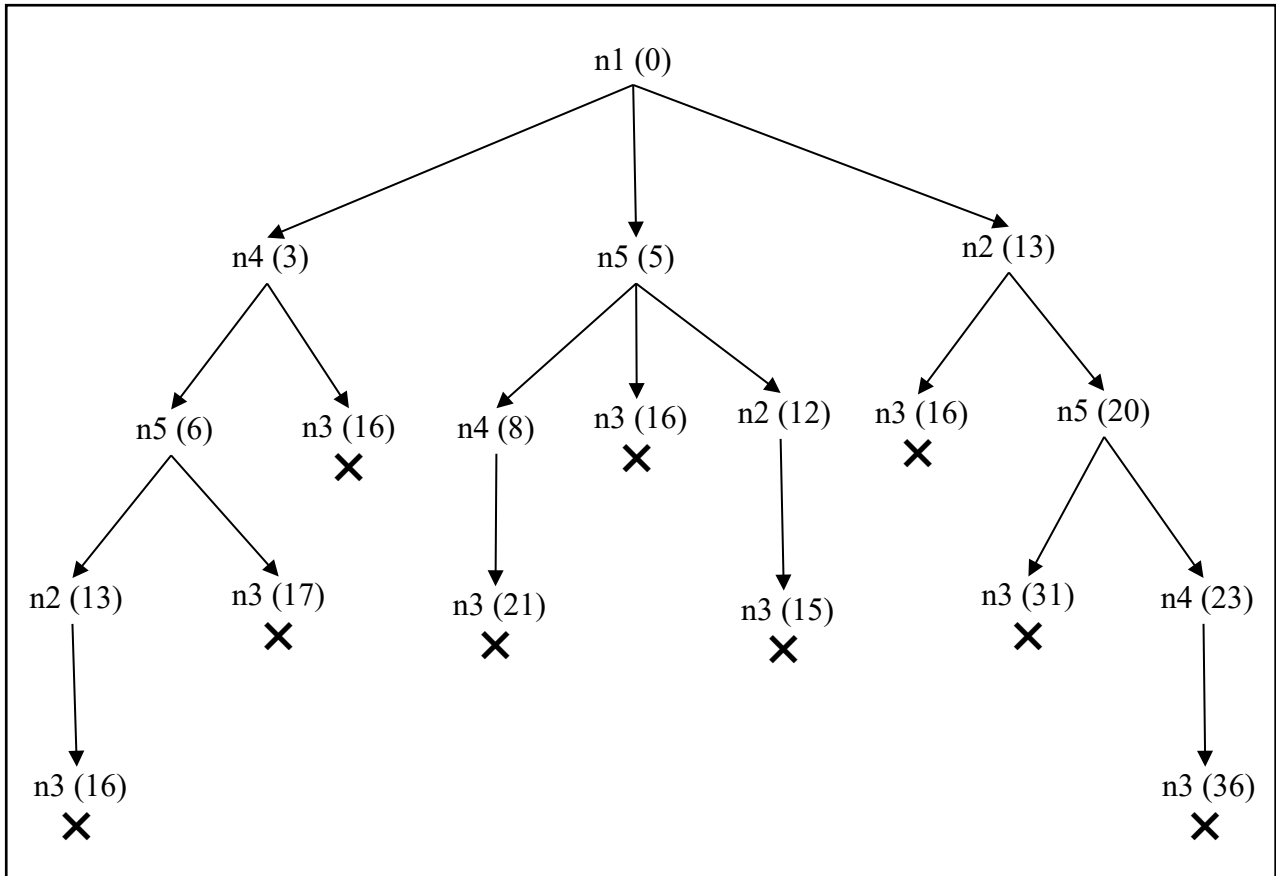
Per risolvere il problema occorre elencare i cammini semplici tra n1 e n3 (con la loro lunghezza) in maniera *sistematica*, in modo da essere certi di averli presi in esame *tutti*. Da tale elenco la soluzione segue immediatamente.

[n1, n2, n3]	16
[n1, n2, n5, n3]	31
[n1, n2, n5, n4, n3]	36
[n1, n5, n3]	16
[n1, n5, n2, n3]	15
[n1, n5, n4, n3]	21
[n1, n4, n3]	16
[n1, n4, n5, n3]	17
[n1, n4, n5, n2, n3]	16

Si noti che, a partire dal nodo n1, si “visitano” i tre nodi adiacenti (n2, n5, n4); a partire da ciascuno di questi si costruiscono tutti i cammini che arrivano a n3 senza passare per un nodo già visitato (cammini semplici).

Una maniera grafica di chiara evidenza (ma anche concettualmente profonda) è illustrata dalla seguente figura che mostra un albero in cui la radice è il nodo di partenza (n1) del grafo, e ogni nodo dell’albero ha tanti figli quanti sono i nodi del grafo a lui collegati purché non compaiono come antenati (nell’albero). Le foglie dell’albero sono il nodo di arrivo (n3) (o un nodo da cui non ci si può più muovere perché il nodo successivo sarebbe un antenato). Ad ogni nodo (dell’albero) è stata aggiunta tra parentesi la distanza dalla radice.

Le foglie che individuano i cammini richiesti sono segnate da una **X** (in questo caso tutte).



PROBLEMA 3

L’ufficio tecnico di un piccolo comune deve scegliere dove piazzare dei nuovi lampioni.

Il paese di cui si parla può essere pensato come un insieme di piazzette collegate da strade, descritte dal seguente grafo (dove i nodi sono le piazze e gli archi sono le strade):

- arco(n1,n2) arco(n2,n3) arco(n3,n1)
- arco(n4,n1) arco(n4,n2) arco(n4,n5)

Ogni lampione illumina la piazza in cui è collocato, le strade da essa uscenti, e le piazze direttamente collegate alla piazza in cui si trova il lampione.

Trovare:

- il numero minimo di lampioni N che consente di illuminare tutte le piazze
- la lista di N piazze su cui collocare i lampioni in modo da illuminare tutte le piazze del paese.

N.B. la lista deve avere gli elementi in ordine crescente (n1 < n2 < ... < n5).

Scrivere la risposta nella seguente tabella.

numero minimo di lampioni	
---------------------------	--

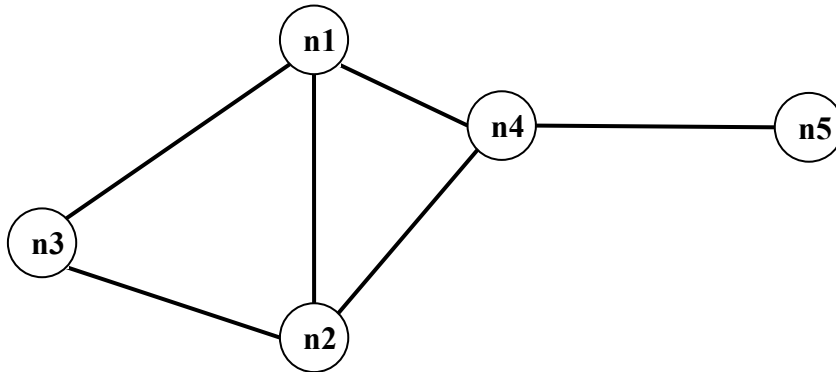
lista delle piazze	
--------------------	--

SOLUZIONE

numero minimo di lampioni	2
lista delle piazze	[n3,n4]

COMMENTI ALLA SOLUZIONE

Il grafo è il seguente.



Un metodo risolutivo generale è: generare tutti i sottoinsiemi di vertici, e per ciascuno verificare se permette di illuminare tutte le piazze del paese; tra tutti quelli che soddisfano tale requisito, prendere quello/quelli con minor numero di elementi.

Questo metodo, può rapidamente diventare impraticabile: bastano già 5 nodi a renderlo difficoltoso. Tuttavia, poiché occorre individuare un sottoinsieme con minor numero di elementi che soddisfa il requisito, è ovvio che conviene generare ed esaminare i sottoinsiemi per cardinalità crescente. Il metodo quindi diventa:

1. verificare se esiste un vertice che da solo illumina tutte le piazze del paese;
 2. generare tutte le coppie e verificare se ne esiste una che illumina tutte le piazze del paese;
 3. generare tutte le triple e verificare se ne esiste una che illumina tutte le piazze del paese;
- etc

Inoltre, in molti casi, semplici osservazioni sulla *topologia* del grafo possono portare immediatamente alla soluzione. Nel caso in esame, è chiaro che per coprire n5, la soluzione deve contenere o n4 oppure n5. Converrà senz'altro provare a costruire la soluzione partendo da n4, e una volta verificato che n4 copre tutti i vertici tranne n3, si arriva subito ad una soluzione minimale.

In generale conviene considerare con particolare attenzione i nodi con valenza maggiore.

PROBLEMA4

Un commesso viaggiatore deve visitare un insieme di città, tornando nel punto di partenza e senza passare due volte per la stessa città (ovvero facendo un *tour*).

Le distanze tra le coppie di città, in chilometri, sono date dai seguenti termini (che hanno la struttura arco(<nome di città>,<nome di città>,<distanza>):

arco(C1,C2,10) arco(C1,C3,4) arco(C1,C4,9)
 arco(C2,C3,6) arco(C2,C4,5) arco(C3,C4,4)

Il commesso parte dalla città C1; (disegnare il grafo delle città e) trovare la lunghezza, in chilometri del *tour* più corto.

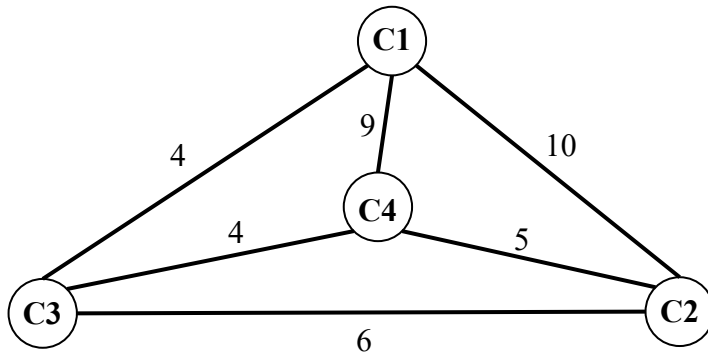
Lunghezza del <i>tour</i> più corto	
-------------------------------------	--

SOLUZIONE

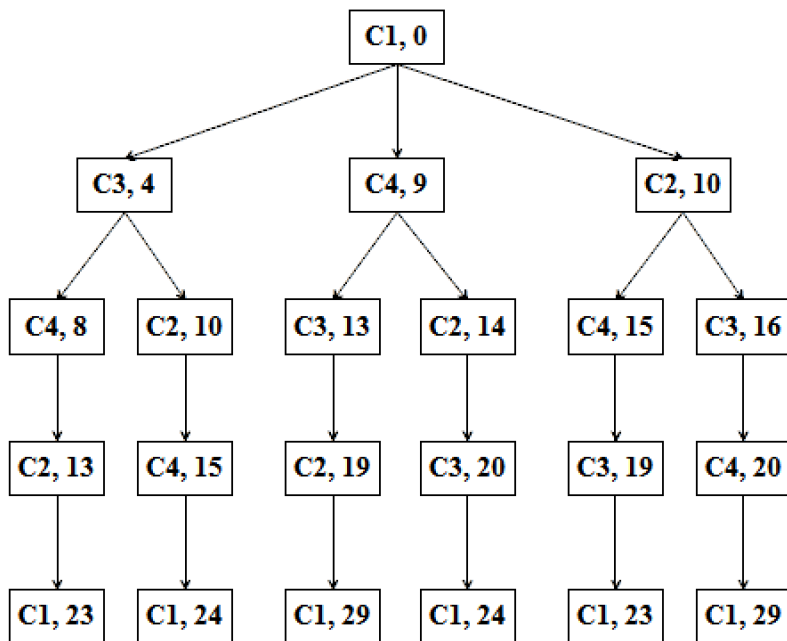
Lunghezza del <i>tour</i> più corto	23
-------------------------------------	----

COMMENTI ALLA SOLUZIONE

Il grafo è il seguente.



Un metodo risolutivo generale è costruire l'albero dei percorsi, da C1 a C1, che toccano una sola volta tutte le città, come nella seguente figura in cui, in ogni nodo è stata riportata la distanza dalla radice.



Si vede che la lunghezza minima di un *tour* è 23 km: tale lunghezza (come tutte le altre) compare due volte perché ogni *tour* ha un suo “simile” percorso in senso inverso.

c) KNAPSACK**PROBLEMA 1**

In un deposito di minerali esistono esemplari di vario peso e valore individuati da sigle di riconoscimento. Ciascun minerale è descritto da una sigla che contiene le seguenti informazioni:

tab(<sigla del minerale>, <valore in euro>, <peso in Kg>).

Il deposito contiene i seguenti minerali:

tab(m1,15,35) tab(m2,19,46) tab(m3,14,25) tab(m4,10,12)

Disponendo di un piccolo motocarro con portata massima di 59 Kg trovare la lista L delle sigle di due minerali diversi che siano trasportabili contemporaneamente con questo mezzo e che abbiano il massimo valore complessivo; calcolare inoltre questo valore V.

N.B. Nella lista, elencare le sigle in ordine (lessicale) crescente; per le sigle usate si ha il seguente ordine: $m1 < m2 < m3 < \dots$

L	
V	

SOLUZIONE

L	[m2,m4]
V	29

COMMENTI ALLA SOLUZIONE

Per risolvere il problema occorre considerare *tutte* le possibili *combinazioni* di due minerali diversi, il loro valore e il loro peso.

N.B. Le *combinazioni* corrispondono ai sottoinsiemi: cioè sono indipendenti dall'ordine; per esempio la combinazione "m1, m4" è uguale alla combinazione "m4, m1". Quindi per elencarle tutte (una sola volta) conviene costruirle sotto forma di liste i cui elementi sono ordinati, come richiesto dal problema: si veda di seguito.

Costruite le combinazioni occorre individuare quelle trasportabili (cioè con peso complessivo minore o eguale a 59) e tra queste scegliere quella di maggior valore.

COMBINAZIONI	VALORE	PESO	TRASPORTABILI
[m1,m2]	15+19=34	35+46=81	no
[m1,m3]	15+14=29	35+25=60	no
[m1,m4]	15+10=25	35+12=47	si
[m2,m3]	19+14=33	46+25=71	no
[m2,m4]	19+10=29	46+12=58	si
[m3,m4]	14+10=24	25+12=37	si

Dal precedente prospetto la soluzione si deduce facilmente.

N.B. Conviene elencare (costruire) prima tutte le combinazioni che iniziano col "primo" minerale, poi tutte quelle che iniziano col "secondo" minerale, e così via, in modo da essere sicuri di averle considerate tutte.

PROBLEMA 2

In un deposito di minerali esistono esemplari di vario peso e valore individuati da sigle di riconoscimento. Ciascun minerale è descritto da un termine che contiene le seguenti informazioni:

minerale(<sigla minerale >,<valore>,<peso>).

Il deposito contiene i seguenti minerali:

minerale(m1,39,58)	minerale(m2,42,64)	minerale(m3,40,65)
minerale(m4,38,59)	minerale(m5,37,61)	minerale(m6,42,62)

I minerali possono essere spostati con carrelli di diversa portata su cui si possono mettere tre esemplari (diversi).

- Disponendo di un carrello con portata massima di 180 Kg, trovare la lista L1 delle sigle di tre minerali diversi che siano trasportabili contemporaneamente e che abbiano il massimo valore complessivo.
- Disponendo di un carrello con portata massima di 185 Kg, trovare la lista L2 delle sigle di tre minerali diversi che siano trasportabili contemporaneamente e che abbiano il massimo valore complessivo.
- Disponendo di un carrello con portata massima di 200 Kg, trovare la lista L3 delle sigle di tre minerali diversi che siano trasportabili contemporaneamente e che abbiano il massimo valore complessivo.

N.B. Nella lista, elencare le sigle in ordine (lessicale) crescente; per le sigle usate si ha il seguente ordine: m1<m2<m3< ...

L1	
L2	
L3	

SOLUZIONE

L1	[m1,m4,m6]
L2	[m1,m2,m6]
L3	[m2,m3,m6]

COMMENTI ALLA SOLUZIONE

In generale, in problemi di questo tipo, occorre considerare *tutte* le possibili *combinazioni* di tre minerali diversi; in questo caso occorre inoltre, per ognuna, determinare il valore, il peso e il carrello più “piccolo” che la può trasportare.

N.B. Le *combinazioni* corrispondono ai sottoinsiemi: cioè sono indipendenti dall’ordine; per esempio la combinazione “m1, m2, m3” è uguale alla combinazione “m3, m2, m1”. Quindi per elencarle tutte (una sola volta) conviene costruirle sotto forma di liste i cui elementi sono ordinati come richiesto dal problema.

COMBINAZIONE	VALORE	PESO	CARRELLO MIN.
[m1,m2,m3]	121	187	200
[m1,m2,m4]	119	181	185
[m1,m2,m5]	118	183	185
[m1,m2,m6]	123	184	185
[m1,m3,m4]	117	182	185
[m1,m3,m5]	116	184	185
[m1,m3,m6]	121	185	185
[m1,m4,m5]	114	178	180

L2

[m1,m4,m6]	119	179	180	L1
[m1,m5,m6]	118	181	185	
[m2,m3,m4]	120	188	200	
[m2,m3,m5]	119	190	200	
[m2,m3,m6]	124	191	200	L3
[m2,m4,m5]	117	184	185	
[m2,m4,m6]	122	185	185	
[m2,m5,m6]	121	187	200	
[m3,m4,m5]	115	185	185	
[m3,m4,m6]	120	186	200	
[m3,m5,m6]	119	188	200	
[m4,m5,m6]	117	182	185	

Costruite le combinazioni, occorre individuare, per ogni carrello, quella di maggior valore.

d) PIANIFICAZIONE**PROBLEMA 1**

Alcuni ragazzi decidono di costruire un ipertesto multimediale sugli avvenimenti significativi della loro regione per la prossima stagione turistica. Per organizzare il progetto, dividono il lavoro in singole attività e, per ciascuna di queste stabiliscono quanti di loro devono partecipare e stimano il tempo per portarla a conclusione. La tabella che segue descrive le attività (indicate rispettivamente con le sigle A1, A2, A3, ...), riportando per ciascuna di esse il numero di ragazzi assegnato e il numero di giorni necessari per completarla.

ATTIVITÀ	RAGAZZI	GIORNI
A1	6	2
A2	4	2
A3	3	3
A4	6	2
A5	4	2
A6	5	1

N.B. Ai fini del problema non è importante conoscere la descrizione delle singole attività.

Le attività devono *succedersi opportunamente* nel tempo perché, per esempio, una attività utilizza il prodotto di altre: quindi esistono delle *priorità* descritte con coppie di sigle; ogni coppia esprime il fatto che l'attività associata alla sigla di destra (detta *successiva*) può iniziare solo quando l'attività associata alla sigla di sinistra (detta *precedente*) è terminata. Ovviamente se una attività ha più precedenti, può iniziare solo quando tutte le precedenti sono terminate.

In questo caso le priorità sono:

[A1,A2], [A1,A3], [A2,A4], [A3,A4], [A4,A5], [A5,A6].

Trovare il numero N di giorni necessari per completare il progetto, tenuto presente che alcune attività possono essere svolte in parallelo e che ogni attività *deve* iniziare prima possibile (nel rispetto delle priorità). Inoltre, trovare il numero massimo RM di ragazzi che lavora contemporaneamente al progetto.

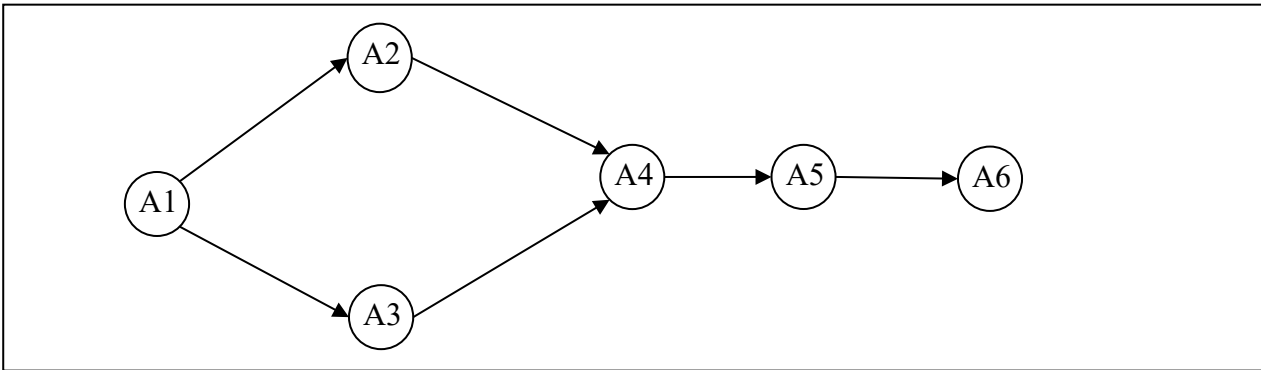
N	
RM	

SOLUZIONE

N	10
RM	7

COMMENTI ALLA SOLUZIONE

Per prima cosa, dai dati sulle priorità occorre disegnare il *diagramma delle precedenze*, cioè il grafo che ha come nodi le attività e come frecce le precedenze: indica visivamente la dipendenza “logica” tra le attività, quindi come si devono susseguire nel tempo.



Per costruire tale grafo (mostrato in figura) si disegnano tanti nodi quante sono le attività (ciascun nodo porta il nome della corrispondente attività).

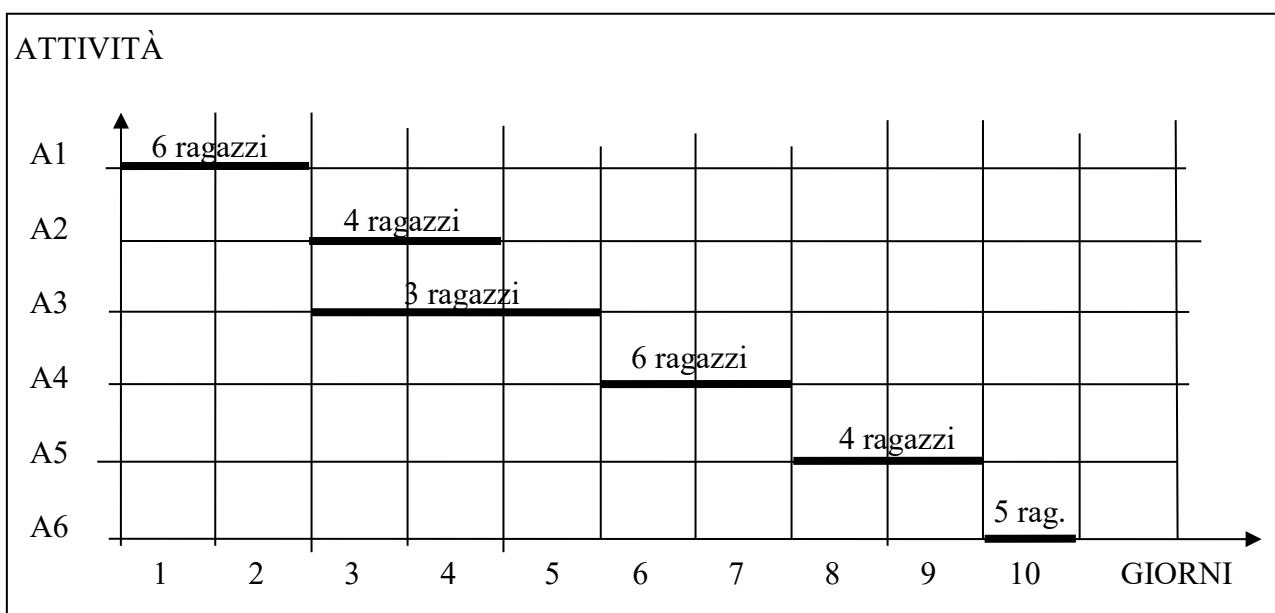
Esiste una attività che compare solo a sinistra nelle coppie che descrivono le priorità: questa è l'attività *iniziale* (in questo caso A1); il nodo corrispondente deve essere disegnato alla sinistra di tutti gli altri.

Esiste una attività che compare solo a destra nelle coppie che descrivono le priorità: questa è l'attività *finale* (in questo caso A6); il nodo corrispondente deve essere disegnato alla destra di tutti gli altri.

Poi per ogni coppia che descrive le priorità si disegna una freccia che connette i nodi coinvolti in quella coppia. Alla fine, in generale, si otterrà un grafo con frecce che si incrociano: tenendo fissi il nodo iniziale e il nodo finale si spostano gli altri nodi per cercare di ottenere (se possibile) un grafo con frecce che non si incrociano (come, appunto, è mostrato in figura).

Poi dal grafo e dalla tabella che descrive le attività, si può compilare il diagramma di Gantt; questo riporta sull'asse verticale le attività (dall'alto verso il basso), sugli assi orizzontali il tempo, in questo caso misurato in giorni. Su ogni asse orizzontale (parallelo a quello dei tempi e in corrispondenza a una attività) è sistemato un segmento che indica l'inizio e la durata della corrispondente attività (e il numero di ragazzi che devono svolgerla).

Così, per esempio, l'attività A1 inizia il giorno 1 e dura due giorni; quando è terminata, il giorno 3 possono iniziare le attività A2 e A3 (che quindi si svolgono parzialmente in parallelo). L'attività A4 può iniziare solamente quando è terminata sia A3 sia A2.



Dal Gantt si vede che il progetto dura 10 giorni e che il numero massimo di ragazzi al lavoro contemporaneamente è 7 (i giorni 3 e 4).

PROBLEMA 2

La tabella che segue descrive le attività di un progetto (indicate rispettivamente con le sigle A1, A2, ...), riportando per ciascuna di esse il numero di persone assegnato e il numero di giorni necessari per completarla.

ATTIVITÀ	PERSONE	GIORNI
A1	6	2
A2	3	3
A3	2	4
A4	6	1
A5	2	3
A6	2	4
A7	3	2
A8	2	4
A9	5	1

Le priorità tra le attività sono:

[A1,A2], [A1,A3], [A1,A4], [A2,A5], [A3,A8], [A7,A9],
[A4,A6], [A6,A7], [A5,A7], [A6,A8], [A8,A9].

Trovare il numero N di giorni necessari per completare il progetto, tenuto presente che alcune attività possono essere svolte in parallelo e che ogni attività *deve* iniziare prima possibile (nel rispetto delle priorità). Inoltre, determinare PM: il *numero massimo* di persone che lavorano contemporaneamente al progetto.

(N.B. PM è anche il *numero minimo* di persone contemporaneamente disponibili necessarie per attuare il progetto così pianificato).

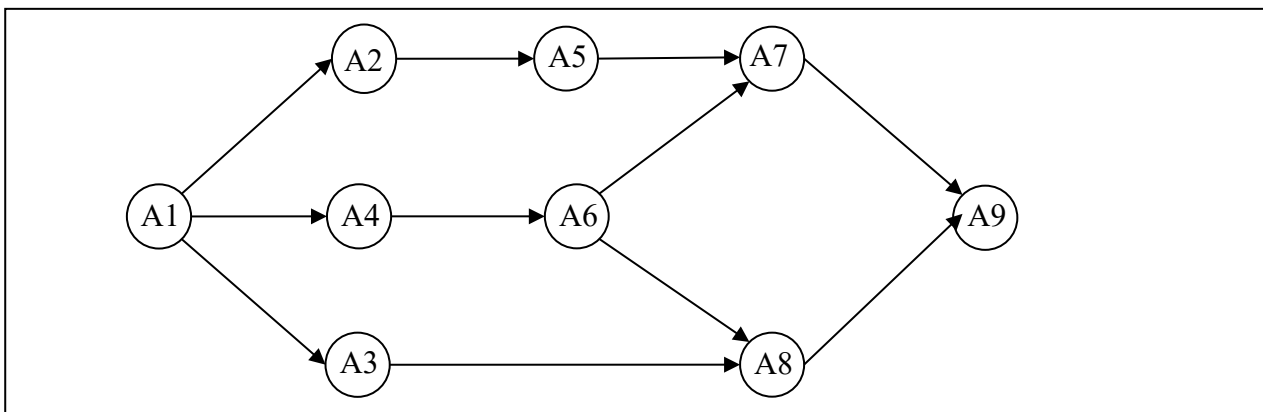
N	
PM	

SOLUZIONE

N	12
PM	11

COMMENTI ALLA SOLUZIONE

Per prima cosa, dai dati sulle priorità occorre disegnare il *diagramma delle precedenze*, cioè il grafo che ha come nodi le attività e come frecce le precedenze.



Tale grafo indica visivamente la dipendenza “logica” tra le attività, quindi come si devono susseguire nel tempo; per costruirlo (come mostrato in figura) si disegnano tanti nodi quante sono le attività (ciascun nodo porta il nome della corrispondente attività).

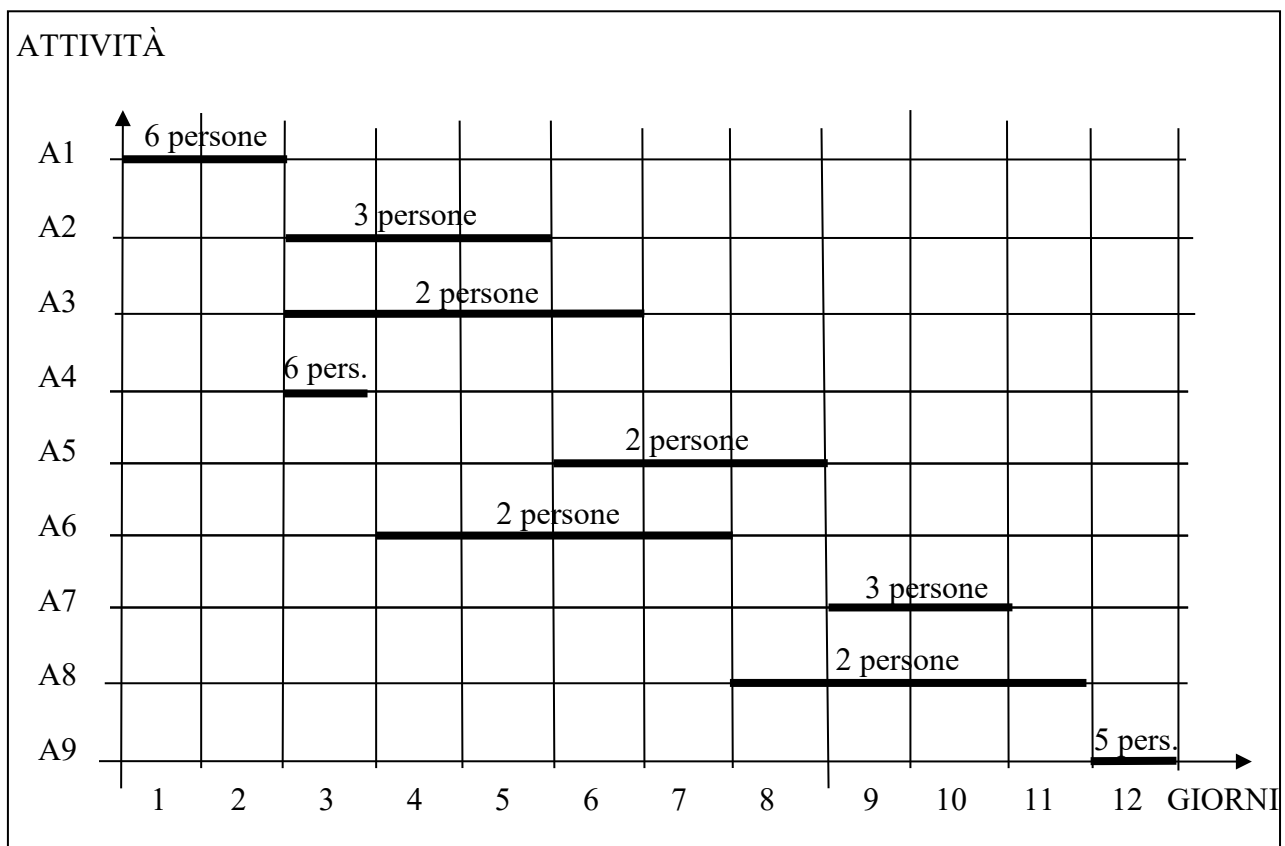
Esiste una attività che compare solo a sinistra nelle coppie che descrivono le priorità: questa è l’attività *iniziale* (in questo caso A1); il nodo corrispondente deve essere disegnato alla sinistra di tutti gli altri.

Esiste una attività che compare solo a destra nelle coppie che descrivono le priorità: questa è l’attività *finale* (in questo caso A9); il nodo corrispondente deve essere disegnato alla destra di tutti gli altri.

Poi per ogni coppia che descrive le priorità si disegna una freccia che connette i nodi coinvolti in quella coppia. Alla fine, in generale, si otterrà un grafo con frecce che si incrociano: tenendo fissi il nodo iniziale e il nodo finale si spostano gli altri nodi per cercare di ottenere un grafo con frecce che non si incrociano.

Poi dal grafo e dalla tabella che descrive le attività, si può compilare il diagramma di Gantt; questo riporta sull’asse verticale le attività (dall’alto verso il basso), sull’asse orizzontale il tempo, in questo caso misurato in giorni. Su ogni linea orizzontale (parallela all’asse dei tempi e in corrispondenza a una attività) è sistemato un segmento che indica l’inizio e la durata della corrispondente attività (e il numero di persone che devono svolgerla): la posizione di tale segmento deve rispettare il diagramma delle precedenze.

Così, per esempio, l’attività A1 inizia il giorno 1 e dura due giorni; quando è terminata, il giorno 3 possono iniziare le attività A2, A3 e A4 (che quindi si svolgono parzialmente in parallelo); inoltre l’attività A7, come altro esempio, può iniziare solamente quando è terminata sia la A5, sia la A6.



Dal Gantt si vede che il progetto dura 12 giorni e che il numero *massimo* di persone al lavoro contemporaneamente è 11 (il terzo giorno): quindi per realizzare il progetto occorre almeno la disponibilità contemporanea di 11 persone.

e) STATISTICA DESCRITTIVA ELEMENTARE**PREMESSA**

Sia dato un insieme di n numeri. Per calcolare la *mediana* dell'insieme:

1. si ordinano i numeri (in maniera crescente o decrescente);
2. se n è dispari, la mediana è il (valore del) numero centrale (cioè quello di posto $(n+1)/2$);
3. se n è pari, la mediana è stimata utilizzando i due valori che occupano le posizione $n/2$ e $(n/2)+1$: generalmente si sceglie la loro media aritmetica (per questa, si veda la definizione più in basso).

Per esempio, la mediana dei numeri presenti nella lista (di 7 elementi) [3,4,6,8,7,5,3] è 5; infatti, nella lista ci sono tre elementi minori di 5 (che sono [3,3,4]) e ce ne sono tre maggiori di 5 (che sono [6,7,8]); la mediana degli elementi della lista [2,3,5,7,2,6,3,7,2,5] si può stimare in 4: anche in questo caso nella lista ci sono cinque elementi più piccoli e un egual numero di elementi più grandi della mediana.

La *moda* di un insieme di numeri è l'elemento ripetuto più volte. Esempio, la moda dell'insieme dei numeri presenti nella lista [1,5,4,2,5,1,4,5] è il numero 5 (ripetuto 3 volte).

La *media (aritmetica)* di n numeri (con $n > 0$) è la loro somma divisa per n .

PROBLEMA

È data la seguente lista di numeri interi:

[1,10,1,11,12]

Trovare la mediana M1.

Trovare la media M2 senza decimali (troncata, non arrotondata).

Trovare la moda M3.

M1	
M2	
M3	

SOLUZIONE

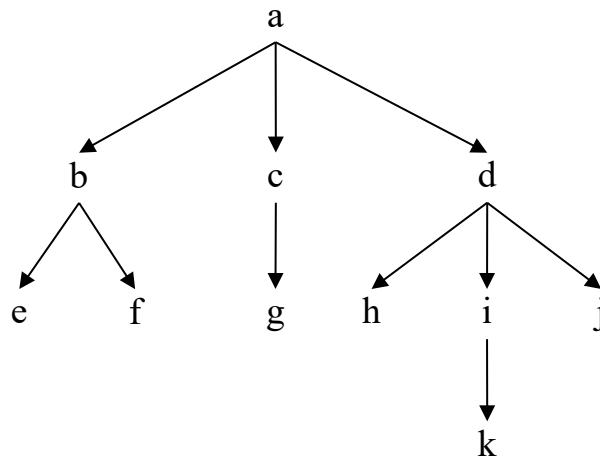
M1	10
M2	7
M3	1

COMMENTI ALLA SOLUZIONE

I risultati seguono immediatamente dalle definizioni.

f) RELAZIONI TRA ELEMENTI DI UN ALBERO**PREMESSA**

La seguente figura rappresenta un albero genealogico che contiene i *nodi* a, b, c, d, e, f, g, h, i, j, k.



Gli alberi di questo tipo possono essere descritti con un insieme di termini del tipo:

arco(<genitore>, <figlio>)

In tal modo, l'albero sopra riportato è descritto dal seguente insieme di termini:

arco(b,e)	arco(b,f)	arco(a,b)	arco(a,c)	arco(c,g)
arco(a,d)	arco(d,h)	arco(d,i)	arco(d,j)	arco(i,k)

Si ricordino i gradi di parentela: gli zii sono i fratelli del genitore, i cugini sono i figli degli zii, il nonno è il padre del padre, ecc. Pertanto, in questo albero:

- il nodo a è nonno di 6 nipoti [e, f, g, h, i, j],
- il nodo k ha 2 zii [h, j],
- il nodo h ha 2 fratelli [i, j] e 3 cugini [e, f, g].

Il nodo a, che non ha genitore, si dice *radice* dell'albero; i nodi [e, f, g, h, j, k] che non hanno figli, si dicono *foglie* dell'albero.

PROBLEMA

Disegnare l'albero genealogico (con radice h) descritto dai seguenti termini:

arco(i,a)	arco(g,b)	arco(g,f)	arco(e,d)
arco(d,c)	arco(h,i)	arco(h,g)	arco(h,e)

Rispondere ai quesiti sottoriportati.

Trovare la lista L1 delle foglie dell'albero, scritte in ordine alfabetico.

Trovare la lista L2 degli zii di b, riportati in ordine alfabetico.

Trovare la lista L3 dei cugini di f, riportati in ordine alfabetico.

Trovare la lista L4 dei nonni presenti nell'albero, riportati in ordine alfabetico.

L1	
L2	
L3	
L4	

SOLUZIONE

L1	[a,b,c,f]
L2	[e,i]

L3	[a,d]
L4	[e,h]

COMMENTI ALLA SOLUZIONE

I risultati seguono immediatamente dalle definizioni.

g) FLUSSI IN UNA RETE DI CANALI**PREMESSA**

Sul fianco di una montagna esistono numerose sorgenti. L'acqua di una sorgente, che si suppone fluire in modo continuo e costante, può scorrere a valle attraverso uno o più canali. Può avvenire che uno o più canali convergano in un punto in cui esiste una sorgente; in tal caso, la loro acqua si aggiunge a quella fornita dalla sorgente raggiunta. Questa situazione è descrivibile con un reticolo di nodi (le sorgenti) collegati da archi (i canali). La situazione quindi è descritta da due tabelle:

$$s(\langle \text{sorgente} \rangle, \langle \text{litri d'acqua erogata al minuto} \rangle),$$

che specifica la quantità d'acqua che sgorga da ogni sorgente (che è un nodo del reticolo),

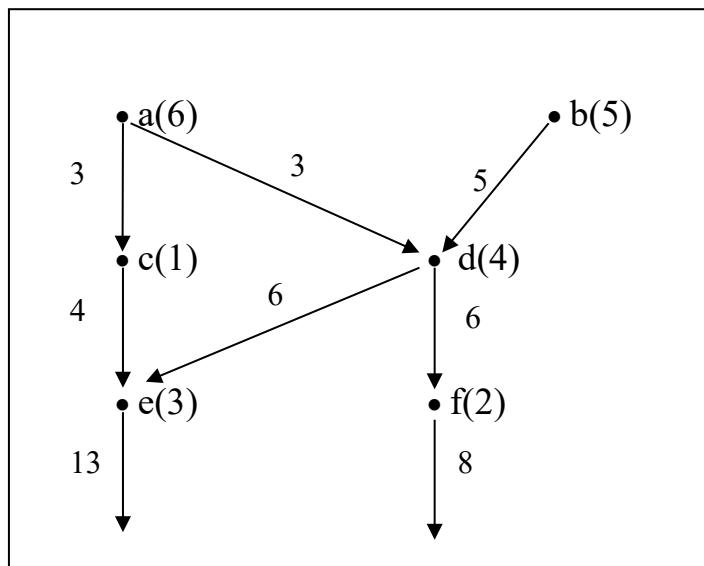
$$r(\langle \text{sorgente1} \rangle, \langle \text{sorgente2} \rangle),$$

che specifica la presenza di un canale che porta acqua dalla sorgente1 alla sorgente2.

Se da una sorgente escono più canali, l'acqua si divide in parti uguali fra ciascuno di essi.

Nella situazione descritta dal seguente esempio (con radici in a e in b, vedi figura)

$$s(a,6), s(b,5), s(c,1), s(d,4), s(e,3), s(f,2)$$

$$r(a,c), r(a,d), r(b,d), r(c,e), r(d,e), r(d,f)$$


la quantità d'acqua che esce dai nodi c, e, f è riportata di seguito.

c	4
e	13
f	8

PROBLEMA

Un reticolo di canali si può descrivere con due tabelle associate ai seguenti due termini:

$$s(\langle \text{sorgente} \rangle, \langle \text{litri d'acqua erogata al minuto} \rangle),$$

che specifica la quantità d'acqua che sgorga da ogni sorgente (che è un nodo del reticolo), e

$$r(\langle \text{sorgente1} \rangle, \langle \text{sorgente2} \rangle),$$

che specifica la presenza di un canale che porta acqua dalla sorgente1 alla sorgente2.

Un reticolo è descritto dalle seguenti due tabelle:

$$s(a,3), s(b,3), s(c,4), s(d,6), s(e,6), s(f,6), s(g,6), s(h,4), s(i,6), s(j,7), s(k,5), s(m,9)$$

$$r(a,e), r(b,g), r(c,e), r(d,g), r(d,h), r(i,b), r(i,d), r(d,f), r(d,e), r(k,d), r(j,d), r(m,d), r(m,c), r(m,a)$$

Disegnare il reticolo, evitando incroci tra i rigagnoli, e determinare la quantità di acqua che esce dai nodi e, f, g, h.

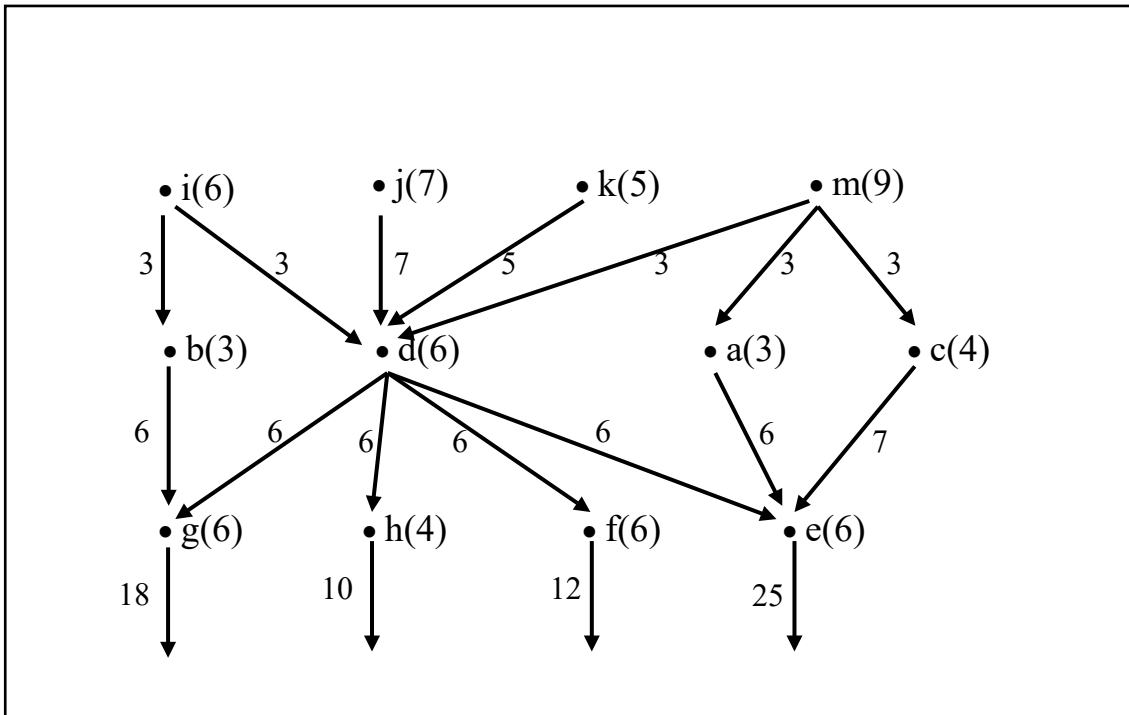
e	
f	
g	
h	

SOLUZIONE

e	25
f	12
g	18
h	10

COMMENTI ALLA SOLUZIONE

Occorre essenzialmente disegnare il reticolo; la portata delle sorgenti è assegnata; la soluzione segue applicando le regole per calcolare la portata dei canali. Naturalmente occorre aggiungere dei canali in uscita dai nodi g, h, f, e.



h) CRITTOGRAFIA**PREMESSA**

Per crittografare un messaggio si può usare una regola per sostituire ogni lettera del messaggio con un'altra. La tabella sotto riportata (che si riferisce all'alfabeto latino di 26 lettere) rappresenta un esempio di regola detta di Giulio Cesare.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
5	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
2	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
20	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t

Con questa regola ogni lettera del messaggio originale, letta nella prima riga, viene sostituita da quella che le corrisponde, per esempio, nella seconda riga; questa è ottenuta “ruotando” l'alfabeto di 5 posizioni più avanti: questo 5 viene detto chiave di cifratura (di Giulio Cesare). Le altre righe sono ottenute ruotando l'alfabeto di una quantità pari al numero posto in prima colonna. Se si scrivono come liste sia il messaggio “in chiaro” sia il corrispondente crittografato, nella tabella seguente sono riportati alcuni esempi di crittografia.

LISTA ORIGINALE	CHIAVE	LISTA CRITTOGRAFATA
[n,a,p,o,l,i]	5	[s,f,u,t,q,n]
[r,o,m,a]	2	[t,q,o,c]
[r,o,m,a]	20	[l,i,g,u]

PROBLEMA

Usando la semplice crittografia di Giulio Cesare:

data la lista [m,i,l,a,n,o] trovarne la corrispondente L1 crittografata con chiave 3;

data la lista [b,o,l,o,g,n,a] trovarne la corrispondente L2 crittografata con chiave 4;

data la lista [w,j,g,j,b,i,v] trovarne la corrispondente L3 crittografata con chiave 5;

L1	
L2	
L3	

SOLUZIONE

L1	[p,l,o,d,q,r]
L2	[f,s,p,s,k,r,e]
L3	[b,o,l,o,g,n,a]

COMMENTI ALLA SOLUZIONE

È sufficiente compilare la tabella in cui la prima riga è il normale alfabeto e le tre successive siano “ruotate” rispettivamente di 3, 4, 5.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
3	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
4	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
5	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e

i) MOVIMENTI DI UN ROBOT O DI PEZZI DEGLI SCACCHI**PREMESSA – PRIMA PARTE**

In un foglio a quadretti è disegnato un “campo di gara”, per esempio di 14 quadretti in orizzontale e 5 in verticale (vedi figura).

									S					
					P									
→														

Ogni casella può essere individuata da due numeri (interi); per esempio la casella contenente P è individuata da essere nella sesta colonna (da sinistra) e nella terza riga (dal basso): brevemente si dice che ha *coordinate* [6,3]; la prima coordinata (in questo caso 6) si dice *ascissa* e la seconda (in questo caso 3) si dice *ordinata*. Le coordinate della casella contenente S sono [10,4] e di quella contenente la freccia sono [1,1].

La freccia può essere pensata come un robot, in questo caso rivolto verso destra; lo stato del robot può quindi essere individuato da tre “valori”: due per le coordinate della casella che occupa e uno per indicare il suo orientamento. Per quest’ultimo si possono usare i simboli della stella dei venti: E, S, W, N: per indicare che il robot è rivolto, rispettivamente, a *destra*, in *basso*, a *sinistra*, in *alto* (con riferimento a chi guarda il foglio); lo stato del robot, rappresentato dalla freccia nella figura è [1,1,E].

Il robot può eseguire tre tipi di comandi:

- girarsi di 90 gradi in senso *orario*: comando **o**;
- girarsi di 90 gradi in senso *antiorario*: comando **a**;
- avanzare di una casella (nel senso della freccia, mantenendo l’orientamento): comando **f**.

Questi comandi possono essere concatenati in sequenze in modo da permettere al robot di compiere vari percorsi; per esempio la sequenza di comandi descritta dalla lista [f,f,f,f,f,a,f,f] fa spostare il robot dalla posizione e orientamento iniziali mostrati in figura fino alla casella P; le caselle via via occupate (quella di partenza e quella di arrivo comprese) sono quelle della lista:

[[1,1],[2,1],[3,1],[4,1],[5,1],[6,1],[6,2],[6,3]].

Stessa casella di arrivo si raggiunge con la lista di comandi [a,f,f,o,f,f,f,f], ma il percorso è diverso ed è descritto dalla lista

[[1,1],[1,2],[1,3],[2,3],[3,3],[4,3],[5,3],[6,3]].

Inoltre, nel primo caso l’orientamento finale del robot è verso l’alto (stato [6,3,N]), mentre nel secondo caso l’orientamento finale è verso destra (stato [6,3,E]).

PREMESSA – SECONDA PARTE

In un foglio a quadretti è disegnato un campo di gara, per esempio di dimensioni 14×5 (cioè 14 quadretti in orizzontale e 5 in verticale, vedi figura).

		Q												
		5	■	■		■			S					
			7	P										
■	■	3												
↑		■												

1	passo f	[8,9]	verso l'alto
2	passo f	[8,10]	verso l'alto
3	passo o	[8,10]	verso destra
4	passo f	[9,10]	verso destra
5	passo f	[10,10]	verso destra
6	passo a	[10,10]	verso l'alto
7	passo f	[10,11]	verso l'alto
8	passo f	[10,12]	verso l'alto
9	passo f	[10,13]	verso l'alto
10	passo o	[10,13]	verso destra
11	passo f	[11,13]	verso destra

PROBLEMA 2

In un campo di gara il robot (come descritto nella prima parte della premessa) è nella casella [9,9] con orientamento verso sinistra: trovare la lista L dei comandi da assegnare al robot per fargli compiere il percorso descritto dalla seguente lista di caselle [[9,9],[8,9],[7,9],[6,9],[6,8],[6,7],[6,6],[6,5]]

L	
---	--

SOLUZIONE

L	[f,f,f,a,f,f,f,f]
---	-------------------

COMMENTI ALLA SOLUZIONE

Per risolvere il problema è conveniente visualizzare il percorso, come nella figura che segue.

					←	←	←	←
					↓			
					↓			
					↓			
					↓			

Dalla figura è immediato che la sequenza di comandi relativa al percorso è la seguente:

- 1 f
- 2 f
- 3 f
- 4 a
- 5 f
- 6 f
- 7 f
- 8 f

Si noti che il quarto comando fa voltare il robot verso il basso, per dargli l'orientamento opportuno per proseguire il percorso, ma non gli fa cambiare posizione.

PROBLEMA 3

Un campo di gara (come descritto nella seconda parte della premessa) ha dimensioni 5×5; le caselle interdette descritte dalla seguente lista:

[[1,2],[1,3],[1,4],[2,4],[3,4],[4,1],[4,2],[4,4],[4,5],[5,4]];

i premi, invece, sono descritti dalla seguente lista:

[[3,1,10],[2,2,12],[2,3,13]].

Al robot sono *vietati* i movimenti corrispondenti alle direzioni della rosa dei venti indicate nella seguente lista [oso,nno,ene], cioè le mosse del robot in questo problema si riducono a quelle illustrate (col cavallo) nella seguente figura.

	×		♞	
♞				×
		♚		
×				♞
	♞		♞	

Partendo dalla casella [1,1], il robot deve raggiungere la casella [5,5], senza passare più di una volta per una stessa casella. Trovare:

- il percorso L1 in cui si raccoglie il massimo di premi;
- il percorso L2 in cui si raccoglie il minimo di premi;
- il numero N di percorsi possibili da [1,1] a [5,5].

L1	
L2	
N	

SOLUZIONE

L1	[[1,1],[2,3],[3,1],[4,3],[5,5]]
L2	[[1,1],[2,3],[3,5],[4,3],[5,5]]
N	2

COMMENTI ALLA SOLUZIONE

Il campo di gara è mostrato nella figura.

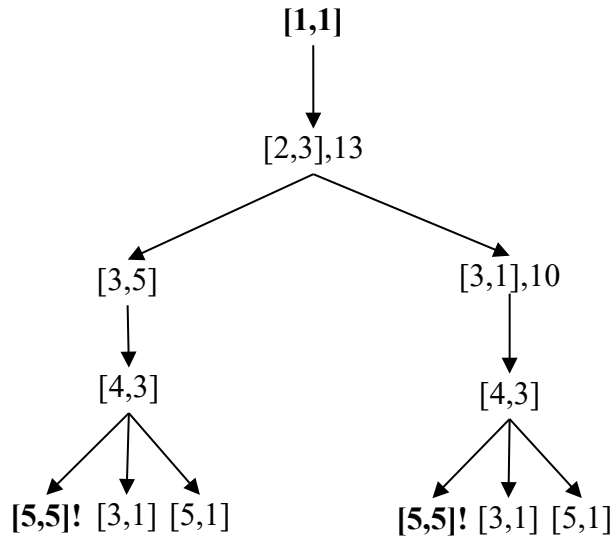
			■	
■	■	■	■	■
■	13			
■	12		■	
♚		10	■	

Esiste una maniera sistematica per trattare problemi di questo tipo: costruire l'*albero delle possibili mosse*. Come mostrato nella seguente figura, si inizia con la *radice* che è la casella in cui parte il robot; poi ad ogni nodo si aggiungono tanti *figli* quante sono le caselle raggiungibili dal robot posto

nella casella corrispondente a quel nodo. Per esempio da [1,1] si può andare in [2,3] (ma non in [3,2] perché è una mossa vietata); poi, per esempio, da [2,3] si può andare in [3,5], [3,1] (ma non in [1,5] perché è una mossa vietata né in [4,4] o [4,2] perché sono caselle interdette) e così via. Nell'albero in figura sono stati indicati i premi a fianco delle caselle nodo, dove appropriato.

N.B. Naturalmente il robot non può tornare in una casella in cui è già stato.

Ci si arresta quando si è arrivati alla *meta* (caratterizzata da un “!” in figura) o in una casella da cui non ci si può muovere.

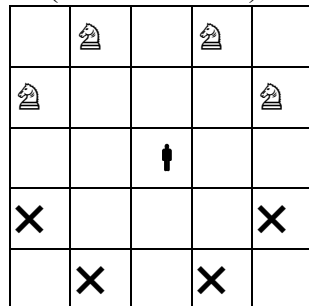


I percorsi sono successioni di nodi dalla radice alle foglie meta: quelli possibili e i premi raccolti sono quindi:

PERCORSO	PREMI RACCOLTI
[[1,1],[2,3],[3,5],[4,3],[5,5]]	13
[[1,1],[2,3],[3,1],[4,3],[5,5]]	23

PROBLEMA 2

In un campo di dimensioni 8×8 (come descritto nella seconda parte della premessa) un robot si muove come il cavallo nel giuoco degli scacchi; gli sono vietate, però, le mosse nelle direzioni della rosa dei venti comprese nella seguente lista [oso,sso,sse,ese], cioè le mosse del robot in questo problema si riducono a quelle illustrate (col simbolo ♞) nella seguente figura.



Nel campo di gara le caselle della seguente lista sono interdette al robot:

[[3,1],[4,4],[4,5],[4,8],[5,2],[5,3],[7,4],[7,5]].

N.B. Un elemento della lista descrive una casella indicandone le coordinate a partire dallo spigolo in basso a sinistra del campo di gara.

Inoltre, in certe caselle sono presenti dei premi, descritti dalla seguente lista:

[[3,2,5],[1,3,6],[2,5,7],[4,6,8],[5,5,9]].

N.B. Un elemento della lista ha la forma: [<ascissa>,<ordinata>,<premio>].

Partendo dalla casella [1,1], il robot deve raggiungere la casella [8,8], senza passare più di una volta per una stessa casella. Trovare:

1. la lista L1 del percorso in cui si raccoglie il minor numero di premi;
2. la lista L2 del percorso in cui si raccoglie il maggior numero di premi.

L1	
L2	

SOLUZIONE

L1	[[1,1],[3,2],[1,3],[3,4],[4,6],[6,7],[8,8]]
L2	[[1,1],[3,2],[1,3],[2,5],[4,6],[6,7],[8,8]]

COMMENTI ALLA SOLUZIONE

Esiste una maniera sistematica per trattare problemi di questo tipo: costruire l'*albero delle possibili mosse*. Ogni nodo dell'albero è etichettato con le coordinate di una casella; si inizia con la *radice* che è la casella in cui parte il robot; poi ad ogni nodo si aggiungono tanti *figli* quante sono le caselle raggiungibili dal robot posto nella casella corrispondente a quel nodo. Naturalmente il robot non può tornare in una casella in cui è già stato.

I nodi in cui ci si arresta (cioè le foglie dell'albero) sono la *meta* o una casella da cui il robot non si può muovere.

In casi "semplici", come il presente, si possono costruire direttamente tutti i percorsi possibili.

N.B. Il presente caso è "semplice" perché al robot è permesso di *muoversi solo verso l'alto*, quindi è facile visualizzarne i percorsi: in particolare all'inizio non potrà "salire" troppo, perché altrimenti non può raggiungere la ottava colonna (appunto perché è limitato nei movimenti).

Il campo di gara è mostrato nella figura.

			■				⌂
			8				
	7		■	9		■	
			■			■	
6				■			
		5		■			
⤴		■					

È facile convincersi che da [1,1] il robot non può andare in [2,3] perché, essendo la casella [4,4] interdetta e dovendo muoversi verso l'alto, non potrà mai raggiungere la ottava colonna (quindi non potrà arrivare in [8,8]).

Quindi la prima mossa è obbligata in [3,2]. Delle 4 possibili mosse successive, due sono impossibili perché in caselle interdette ([5,3] e [4,4]), la terza in [2,4] non permette al robot di raggiungere la ottava colonna (visto che anche la [4,5] è interdetta). Rimane solo la mossa in [1,3].

La terza mossa può essere in [2,5] oppure in [3,4]. Nel primo caso il robot è già salito “molto” e ha un unico percorso obbligato per arrivare alla meta [8,8]: continuare in [4,6], [6,7] e infine [8,8].

Nel secondo caso è facile vedere che sono possibili solo tre continuazioni:

[4,6], [6,7], [8,8] (come nel caso precedente)

[5,5], [6,7], [8,8]

[5,5], [7,6], [8,8]

Ricapitolando, i percorsi possibili e i premi raccolti sono:

[[1,1],[3,2],[1,3],[2,5],[4,6],[6,7],[8,8]]	26	L2
[[1,1],[3,2],[1,3],[3,4],[4,6],[6,7],[8,8]]	19	L1
[[1,1],[3,2],[1,3],[3,4],[5,5],[6,7],[8,8]]	20	
[[1,1],[3,2],[1,3],[3,4],[5,5],[7,6],[8,8]]	20	

j) SOTTOSEQUENZE**PREMESSA**

Una sequenza può essere pensata come una lista; per esempio la seguente è una sequenza di numeri interi (non necessariamente distinti):

[15,6,12,18,9,8,10,20,8,4,7]

Una *sottosequenza* è una lista che contiene una parte degli elementi di quella originale, posti nello stesso ordine. Esempi di sottosequenze della lista precedente sono:

[15,18,20,4], [15,6,12,18,7], [9,8,10, 8,4,7]

Non è una sottosequenze della lista precedente la seguente:

[15,18,12,9,8,20,10,4,]

perché gli elementi non compaiono nello stesso ordine di quella data (per esempio 18 e 12 oppure 20 e 10).

In certi casi, data una sequenza, è rilevante determinare sottosequenze con certe proprietà; un caso tipico è determinare la sottosequenza più lunga (strettamente) *decescente*, o quella (strettamente) *crescente*, oppure quella i cui elementi godono di certe proprietà.

N.B. “strettamente” indica che non ci sono elementi ripetuti.

PROBLEMA 1

Un bambino ama molto guardare i cartoni animati in TV. I genitori gli impongono la regola che dopo aver guardato un programma di durata d , può guardare solo programmi di durata minore di d . Al bambino viene data la Guida TV del suo canale preferito che contiene l’elenco di tutti i programmi della giornata con le relative durate. Aiutatelo a scegliere il più grande insieme di programmi che può guardare senza infrangere la regola imposta, se la sequenza delle durate dei programmi (esprese in minuti) è la seguente:

[7,20,12,14,13,15,8,4]

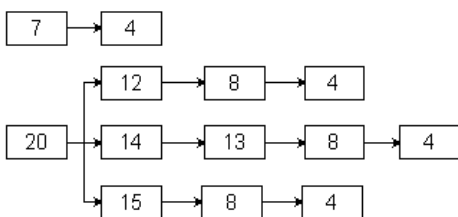
Scrivere la risposta nella casella sottostante.

SOLUZIONE

COMMENTI ALLA SOLUZIONE

Il problema chiede di determinare la *sottosequenza decrescente più lunga* estratta dalla lista data. Nei casi semplici come questo si può procedere “ad occhio”; nei casi più complessi occorre essere sistematici ed esaminare *tutte* le possibili sequenze: a partire da ogni elemento della lista si costruisce, in generale, un albero di sequenze (questo processo si può chiamare “sviluppo dell’elemento”); quindi, si otterrà una “foresta”.

Una osservazione che abbrevia di molto il lavoro è la seguente: è inutile sviluppare un nodo se questo compare nello sviluppo di uno già sviluppato; quindi le sequenze da esaminare (e costruire) sono solo le seguenti:



La soluzione ([20,14,13,8,4]) segue immediatamente.

PROBLEMA 2

Si consideri la sequenza di numeri interi:

[8,10,11,12,4,18,6,7,14,10,18,20].

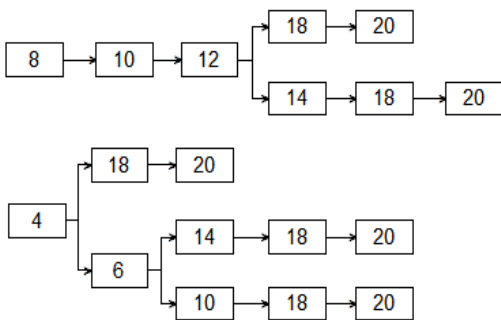
Determinare la più lunga sottosequenza strettamente crescente di numeri pari.

Scrivere la risposta nella casella sottostante.

SOLUZIONE

COMMENTI ALLA SOLUZIONE

Si possono individuare 5 sottosequenze, come mostrato dalla figura che segue.



Tale figura è costituita da alberi che, per comodità disegniamo orizzontali; la radice di un albero è un elemento della lista data che non è già comparso in un albero precedente. Ogni nodo dell'albero ha come figli gli elementi della lista che sono contemporaneamente:

1. a lui successivi, cioè a destra nella lista data (si cercano le sottosequenze),
2. pari (si cercano le sottosequenze di numeri pari),
3. di lui maggiori (si cercano le sottosequenze crescenti),
4. minori dei fratelli (si cercano le sottosequenze più lunghe).

(Per il punto 4 si noti che se l'elemento aggiunto non fosse minore dei fratelli, la sequenza a cui appartiene sarebbe sicuramente più corta di quella a cui appartiene il fratello.)

La soluzione [8,10,12,14,18,20] segue immediatamente.

k) FATTI E CONCLUSIONI**PREMESSA**

Questi problemi trattano di *entità* correlate da fatti; ciascuna entità ha *valori* discreti. Consideriamo, per esempio, le entità “nome”, “cognome”, “età”; se si parla di tre persone, allora il nome può avere (3) valori: Aldo, Giacomo, Giovanni; il cognome può avere i (3) valori Storti, Poretti, Baglio e l’età i (3) valori: 58, 59, 60. Nei problemi vengono enunciati dei fatti e da questi occorre *ragionare* e traendo *conclusioni* per associare opportunamente i valori di nome, cognome ed età.

Per risolvere questi problemi è utile usare un *master board*. Questo si costruisce e si usa mediante i seguenti passi.

1. Si scelgono *due* tra le entità (quelle ritenute “primarie”: comunque la scelta non influenza il risultato finale); in questo caso, per esempio, “nome” e “cognome”.
2. si sistemano le entità in uno schema orizzontale/verticale, come il seguente; nella riga orizzontale si sistemano tutte le entità tranne la seconda delle primarie; nella riga verticale si sistemano tutte le entità tranne la prima delle primarie.
3. N.B. Le entità orizzontali, *a parte la prima*, e quelle verticali, *a parte la prima*, devono essere elencate in ordine *inverso*. In questo semplice caso la condizione non è rilevante perché ci sono solo 3 entità: per un esempio, si veda il successivo PROBLEMA 2.

	cognome	età
nome		
età		

4. si completa lo schema aggiungendo i *quadranti*, come indicato nella seguente figura, *omettendo* i quadranti con le stesse “coordinate”.

	cognome	età
nome		
età		

5. Un incrocio di entità diverse si chiama *quadrante* (o *ambito*): nella figura precedente ci sono tre quadranti (ambiti): nome/cognome, nome/età ed età/cognome. Un quadrante si identifica con le sue “coordinate”: entità orizzontale/entità verticale.
6. Al posto delle entità si riportano i valori, ripartendo i quadranti nel numero opportuno di celle.

	Storti	Poretti	Baglio	58	59	60
Aldo						
Giacomo						
Giovanni						
58						
59						
60						

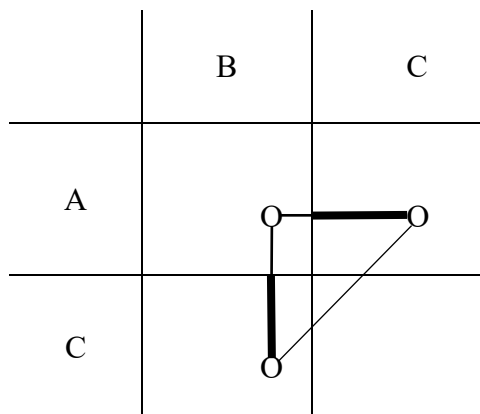
7. Si riportano i “fatti” sul *master board*, esaminandoli uno dopo l’altro, e traducendoli nell’aggiunta di “X” negli “incroci vietati” e di “O” negli “incroci consentiti”.
8. Successivamente si inizia il “ragionamento”: come descritte immediatamente nel seguito, e illustrate negli esempi, si applicano *ripetutamente* (finché è possibile) i seguenti 5 tipi di *conclusioni*.

Quando nessuno dei tipi di conclusione è più applicabile, il “ragionamento” è terminato.

LE CONCLUSIONI

Le *conclusioni* si possono dedurre dai due seguenti principi.

- I. Principio della *non contraddittorietà*: in ogni riga e ogni colonna di un quadrante deve esserci una sola “O” (e le altre caselle sono occupate da “X”).
- II. Principio del *triangolo completo*: dati tre quadranti, A/B, A/C e C/B disposti come nella seguente figura:



se esistono due vertici qualsiasi del triangolo, allora esiste il triangolo *completo* e i segmenti marcati sono eguali (cioè sono eguali i valori della entità C).

N.B. Il quadrante A/C può non essere *immediatamente* a destra di A/B e il quadrante B/C può non essere *immediatamente* sotto A/B.

N.B. In casi in cui il master board è più grande il triangolo può essere disposto in altra maniera, per esempio coll'angolo retto a destra della ipotenusa.

Il primo principio non ha bisogno di giustificazioni; il secondo principio è una delle tante forme di sillogismo ternario: se $x \in A \cap B$ e $x \in B \cap C$ allora $x \in A \cap C$.

Dal primo principio seguono le conclusioni 1 e 2, dal secondo seguono le conclusioni 3, 4 e 5.

1. *aggiunta* di “X”: se in un quadrante compare una “O”, si completano la riga e la colonna (di quel quadrante) con le “X”;
2. *aggiunta* di una “O”: se in un quadrante compare una riga (o una colonna) che contiene tutte “X” tranne una casella, si completa la riga (o la colonna) con una “O”;
3. *incrocio* delle “O”: in un quadrante A/B si aggiunge una “O” in una casella (libera) se in un quadrante a destra A/C (corrispondente a una certa entità C), c'è una “O” nella stessa riga (con un valore v) e nel quadrante in basso B/C (corrispondente alla *stessa* entità C), c'è una “O” nella stessa colonna della casella libera (con lo stesso valore v);
4. *ribaltamento* delle “O” (conclusione “duale” dell'incrocio):
se in due quadranti sovrapposti (A/B e C/B) c'è una “O” in una stessa colonna, cioè se:
 - in un quadrante A/B compare una “O” in una casella c (individuata da riga e colonna) e
 - in un quadrante in basso C/B (corrispondente a una certa entità C), compare una “O” nella stessa colonna di c (in corrispondenza del valore v dell'entità C)

allora

si aggiunge una “O” nel quadrante A/C a destra di quello dato (corrispondente alla entità C) nella stessa riga di c , in corrispondenza del valore v dell'entità C;

e *viceversa* (scambiando colonna/riga e basso/destra): se in due quadranti affiancati (A/B e A/C) ci sono due “O” nella stessa riga, allora si aggiunge una “O” nella casella opportuna nel quadrante B/C.

5. *non incrocio* (conclusione “complementare” dell'incrocio): in un quadrante A/B si aggiunge una “X” in una casella (libera) se in un quadrante a destra A/C (corrispondente a una certa entità C), c'è una “O” nella stessa riga (con un valore v di C) e nel quadrante in basso C/B (corrispondente alla *stessa* entità C), c'è una “X” nella stessa colonna della casella libera (con lo stesso valore v) e *viceversa* (scambiando colonna/riga e basso/destra): in un quadrante A/B si aggiunge una “X” in una casella (libera) se in un quadrante a destra A/C (corrispondente a una certa entità C), c'è una “X” nella stessa riga (con un valore v) e nel quadrante in basso C/B (corrispondente alla *stessa* entità C), c'è una “O” nella stessa colonna della casella libera (con lo stesso valore v).

N.B. Conclusioni analoghe alle ultime tre si traggono nel caso il triangolo sia disposto in altra maniera.

PROBLEMA 1

Alice, Bastiano e Carla abitano nella stessa strada. I loro cognomi sono Rossi, Verdi e Bianchi, e le loro età sono 17, 19 e 20. I nomi, i cognomi e le età sono elencati in ordine casuale (e quindi non si corrispondono ordinatamente).

Dai due fatti elencati di seguito, determinare il nome completo e l'età di ogni persona, riempiendo la successiva tabella.

1. La signorina Rossi è tre anni più vecchia di Carla.
2. La persona di cognome Bianchi ha 19 anni.

NOMI	COGNOMI	ETÀ
Alice		
Bastiano		
Carla		

SOLUZIONE

NOMI	COGNOMI	ETÀ
Alice	Rossi	20
Bastiano	Bianchi	19
Carla	Verdi	17

COMMENTI ALLA SOLUZIONE

Nell'enunciato del problema compaiono tre entità: nomi, cognomi, età; si può assumere che la coppia primaria sia nome e cognome e quindi il *master board* è il seguente:

	Rossi	Verdi	Bianchi	17	19	20
Alice						
Bastiano						
Carla						
17						
19						
20						

Si riportano i fatti nel *master board*, aggiungendo a pedice il numero del fatto da cui la “O” o la “X” provengono.

Il primo fatto è complesso: infatti equivale ad affermare che Rossi ha 20 anni e Carla ne ha 17 e di conseguenza Rossi non è Carla, però è signorina, quindi non può essere Bastiano: è Alice

Il secondo fatto è semplice e immediatamente riportabile nel *master board*.

	Rossi	Verdi	Bianchi	17	19	20
Alice	O₁					
Bastiano						
Carla				O₁		
17						
19			O₂			
20	O₁					

Si traggono le conclusioni (che consistono nei cinque tipi visti) iterativamente, cioè finché non è più possibile applicarne alcuna; si aggiunge a pedice la lettera che distingue la conclusione da cui la “O” o la “X” provengono.

a) aggiunta di “X” (a completamento delle “O”).

	Rossi	Verdi	Bianchi	17	19	20
Alice	O₁	X_a	X_a	X_a		
Bastiano	X_a			X_a		
Carla	X_a			O₁	X_a	X_a
17	X_a		X_a			
19	X_a	X_a	O₂			
20	O₁	X_a	X_a			

b) aggiunta di “O” (a completamento delle “X”).

	Rossi	Verdi	Bianchi	17	19	20
Alice	O_1	X_a	X_a	X_a		
Bastiano	X_a			X_a		
Carla	X_a			O_1	X_a	X_a
17	X_a	O_b	X_a			
19	X_a	X_a	O_2			
20	O_1	X_a	X_a			

c) ribaltamento di O_1 all'incrocio Rossi/20: Alice che è Rossi ha 20 anni.

	Rossi	Verdi	Bianchi	17	19	20
Alice	O_1	X_a	X_a	X_a		O_c
Bastiano	X_a			X_a		
Carla	X_a			O_1	X_a	X_a
17	X_a	O_b	X_a			
19	X_a	X_a	O_2			
20	O_1	X_a	X_a			

d) aggiunta di X a completamento delle O.

	Rossi	Verdi	Bianchi	17	19	20
Alice	O_1	X_a	X_a	X_a	X_d	O_c
Bastiano	X_a			X_a		X_d
Carla	X_a			O_1	X_a	X_a
17	X_a	O_b	X_a			
19	X_a	X_a	O_2			
20	O_1	X_a	X_a			

e) aggiunta di O a completamento delle X.

	Rossi	Verdi	Bianchi	17	19	20
Alice	O_1	X_a	X_a	X_a	X_a	O_c
Bastiano	X_a			X_a	O_e	X_a
Carla	X_a			O_1	X_a	X_a
17	X_a	O_b	X_a			
19	X_a	X_a	O_2			
20	O_1	X_a	X_a			

f) incrocio delle O: Bastiano ha 19 anni e Bianchi ha 19 anni, quindi sono la stessa persona.

	Rossi	Verdi	Bianchi	17	19	20
Alice	O_1	X_a	X_a	X_a	X_a	O_c
Bastiano	X_a		O_f	X_a	O_e	X_a
Carla	X_a			O_1	X_a	X_a
17	X_a	O_b	X_a			
19	X_a	X_a	O_2			
20	O_1	X_a	X_a			

g) un passo di aggiunta delle “X” e di aggiunta delle “O” concludono la risoluzione.

	Rossi	Verdi	Bianchi	17	19	20
Alice	O_1	X_a	X_a	X_a	X_a	O_c
Bastiano	X_a	X_g	O_f	X_a	O_b	X_a
Carla	X_a	O_g	X_g	O_1	X_a	X_a
17	X_a	O_b	X_a			
19	X_a	X_a	O_2			
20	O_1	X_a	X_a			

N.B. Il processo di soluzione sembra “lungo” perché le figure sono ripetute ed occupano “molto” spazio; ragionando su una (sola) copia del *master board*, il processo consiste in due passi per riportare i fatti e 7 passi di conclusioni, essenzialmente tutti *evidenti e automatici*.

PROBLEMA 2

Alice, Bastiano e Carla sono pronti per i Grandi Giochi Estivi. Vivono in tre grandi città: Roma, Napoli e Milano; si sono allenati duramente: chi per 3 mesi, chi per 5 mesi chi, addirittura per 7 mesi; praticano il golf, il canottaggio e il ciclismo.

Dai fatti elencati di seguito, determinare dove vive ogni atleta, per quanto tempo si è allenato e che sport pratica, riempiendo la successiva tabella.

1. Chi pratica golf si è allenato per 3 mesi.
2. Bastiano, che non si è allenato per 5 mesi, pratica il ciclismo.
3. Alice si è allenata per due mesi più dell'atleta che sta a Roma.
4. Chi pratica canottaggio non sta a Milano.

NOMI	CITTÀ	SPORT	TEMPO
Alice			
Bastiano			
Carla			

SOLUZIONE

NOMI	CITTÀ	SPORT	TEMPO
Alice	Napoli	canottaggio	5
Bastiano	Milano	ciclismo	7
Carla	Roma	golf	3

COMMENTI ALLA SOLUZIONE

Nell'enunciato del problema compaiono quattro entità: nomi, città, sport e tempo; si può assumere che la *coppia principale* sia nome e città si costruisce il *master board*: si noti che sport e tempo sono sistemati, in orizzontale e in verticale, in ordine inverso.

	Roma	Milano	Napoli	golf	canottaggio	ciclismo	3 mesi	5 mesi	7 mesi
Alice									
Bastiano									
Carla									
3 mesi									
5 mesi									
7 mesi									
golf									
canottaggio									
ciclismo									

Si riportano i fatti nel *master board*, aggiungendo a pedice il numero del fatto da cui la "O" o la "X" provengono.

a) Primo fatto: “Chi pratica golf si è allenato per 3 mesi”:

	Roma	Milano	Napoli	golf	canottaggio	ciclismo	3 mesi	5 mesi	7 mesi
Alice									
Bastiano									
Carla									
3 mesi				O₁					
5 mesi									
7 mesi									
golf									
canottaggio									
ciclismo									

b) Secondo fatto: “Bastiano, che non si è allenato per 5 mesi, pratica il ciclismo”:

	Roma	Milano	Napoli	golf	canottaggio	ciclismo	3 mesi	5 mesi	7 mesi
Alice									
Bastiano						O₂		X₂	
Carla									
3 mesi				O₁					
5 mesi									
7 mesi									
golf									
canottaggio									
ciclismo									

c) Terzo fatto: “Alice si è allenata per due mesi più dell’atleta che sta a Roma” (Alice non sta a Roma e non si è allenata per 3 mesi, l’atleta che sta a Roma non si è allenato per 7 mesi):

	Roma	Milano	Napoli	golf	canottaggio	ciclismo	3 mesi	5 mesi	7 mesi
Alice	X₃						X₃		
Bastiano						O₂		X₂	
Carla									
3 mesi				O₁					
5 mesi									
7 mesi	X₃								
golf									
canottaggio									
ciclismo									

d) Quarto fatto: “Chi pratica canottaggio non sta a Milano”.

	Roma	Milano	Napoli	golf	canottaggio	ciclismo	3 mesi	5 mesi	7 mesi
Alice	X₃						X₃		
Bastiano						O₂		X₂	
Carla									
3 mesi				O₁					
5 mesi									
7 mesi	X₃								
golf									
canottaggio		X₄							
ciclismo									

Si traggono le conclusioni (che consistono nei cinque tipi visti) iterativamente, cioè finché non è più possibile applicarne alcuna; si aggiunge a pedice la lettera che distingue la conclusione da cui la “O” o la “X” provengono.

a) Completamento delle O.

	Roma	Milano	Napoli	golf	canottaggio	ciclismo	3 mesi	5 mesi	7 mesi
Alice	X_3					X_a	X_3		
Bastiano				X_a	X_a	O_2		X_2	
Carla						X_a			
3 mesi				O_1	X_a	X_a			
5 mesi				X_a					
7 mesi	X_3			X_a					
golf									
canottaggio		X_4							
ciclismo									

b) Non incrocio: nella casella Alice/golf si aggiunge una “X”.

	Roma	Milano	Napoli	golf	canottaggio	ciclismo	3 mesi	5 mesi	7 mesi
Alice	X_3			X_b		X_a	X_3		
Bastiano				X_a	X_a	O_2		X_2	
Carla						X_a			
3 mesi				O_1	X_a	X_a			
5 mesi				X_a					
7 mesi	X_3			X_a					
golf									
canottaggio		X_4							
ciclismo									

- c) Completamento delle “X”: si mettono “O” nella prima colonna e nella prima riga del quadrante nomi/sport; completamento delle “O”: si mette una “X” nella casella Carla/canottaggio, completando il quadrante.

	Roma	Milano	Napoli	golf	canottaggio	ciclismo	3 mesi	5 mesi	7 mesi
Alice	X₃			X_b	O_c	X_a	X₃		
Bastiano				X_a	X_a	O₂		X₂	
Carla				O_c	X_c	X_a			
3 mesi				O₁	X_a	X_a			
5 mesi				X_a					
7 mesi	X₃			X_a					
golf									
canottaggio		X₄							
ciclismo									

- d) Ribaltamento della “O” della casella 3 mesi/golf (nella casella Carla/3 mesi) e completamento delle “O”.

	Roma	Milano	Napoli	golf	canottaggio	ciclismo	3 mesi	5 mesi	7 mesi
Alice	X₃			X_b	O_c	X_a	X₃		
Bastiano				X_a	X_a	O₂	X_d	X₂	
Carla				O_c	X_c	X_a	O_d	X_d	X_d
3 mesi				O₁	X_a	X_a			
5 mesi				X_a					
7 mesi	X₃			X_a					
golf									
canottaggio		X₄							
ciclismo									

e) A questo punto si può completare il quadrante nomi/mesi

	Roma	Milano	Napoli	golf	canottaggio	ciclismo	3 mesi	5 mesi	7 mesi
Alice	X_3			X_b	O_c	X_a	X_3	O_e	X_e
Bastiano				X_a	X_a	O_2	X_d	X_2	O_e
Carla				O_c	X_c	X_a	O_d	X_d	X_d
3 mesi				O_1	X_a	X_a			
5 mesi				X_a					
7 mesi	X_3			X_a					
golf									
canottaggio		X_4							
ciclismo									

f) Non incrocio: nella casella Alice/Milano si aggiunge una "X".

	Roma	Milano	Napoli	golf	canottaggio	ciclismo	3 mesi	5 mesi	7 mesi
Alice	X_3	X_f		X_b	O_c	X_a	X_3	O_e	X_e
Bastiano				X_a	X_a	O_2	X_d	X_2	O_e
Carla				O_c	X_c	X_a	O_d	X_d	X_d
3 mesi				O_1	X_a	X_a			
5 mesi				X_a					
7 mesi	X_3			X_a					
golf									
canottaggio		X_4							
ciclismo									

g) Completamento delle “X” nella riga Alice/città e completamento della “O” nella colonna nomi/Napoli.

	Roma	Milano	Napoli	golf	canottaggio	ciclismo	3 mesi	5 mesi	7 mesi
Alice	X ₃	X _f	O _g	X _b	O _c	X _a	X ₃	O _e	X _e
Bastiano			X _g	X _a	X _a	O ₂	X _d	X ₂	O _e
Carla			X _g	O _c	X _c	X _a	O _d	X _d	X _d
3 mesi				O ₁	X _a	X _a			
5 mesi				X _a					
7 mesi	X ₃			X _a					
golf									
canottaggio		X ₄							
ciclismo									

h) Non incrocio e completamento: si può mettere una “X” nella casella Bastiano/Roma e completare il quadrante nomi/città.

	Roma	Milano	Napoli	golf	canottaggio	ciclismo	3 mesi	5 mesi	7 mesi
Alice	X ₃	X _f	O _g	X _b	O _c	X _a	X ₃	O _e	X _e
Bastiano	X _h	O _h	X _g	X _a	X _a	O ₂	X _d	X ₂	O _e
Carla	O _h	X _h	X _g	O _c	X _c	X _a	O _d	X _d	X _d
3 mesi				O ₁	X _a	X _a			
5 mesi				X _a					
7 mesi	X ₃			X _a					
golf									
canottaggio		X ₄							
ciclismo									

A questo punto il problema è risolto; è un utile esercizio completare il *master board*:

- si ribaltano le “O” del quadrante nomi/sport nel quadrante sport/città;
- si ribaltano le “O” del quadrante nomi/mesi nel quadrante mesi/città;
- si incrociano le “O” del quadrante mesi/città con quelle del quadrante nomi/sport per metterle nel quadrante mesi/sport;

- si completano tutte le “O”.

N.B. Anche in questo caso il processo di soluzione sembra “lungo” perché le figure sono ripetute ed occupano “molto” spazio; ragionando su una (sola) copia del *master board*, il processo consiste in 4 passi per riportare i fatti e 8 passi di conclusioni (più un ulteriore passo), essenzialmente tutti *evidenti e automatici*.

4. ELEMENTI DI PSEUDOLINGUAGGIO

N.B. Un documento più completo (dal titolo “La programmazione, ovvero esprimersi chiaramente) è a disposizione sul sito delle OPS

PRIMA PARTE

1. INTRODUZIONE: LA METAFORA

In una cassettera ci sono dei cassette individuati dalle lettere A, B, C, D, E, F. In ciascun cassetto ci può essere un foglietto su cui è scritto un numero *intero*. La scrittura

$$C \leftarrow A+B;$$

significa: “*sommare i numeri scritti sui foglietti dei cassette A e B, scrivere il risultato su un nuovo foglietto e inserire questo foglietto nel cassetto C, dopo aver eliminato il foglietto (eventualmente) presente in C*”. Se all’inizio nei foglietti di A e B sono scritti rispettivamente i numeri 12 e 7, a operazione eseguita in C si trova un foglietto su cui è scritto 19.

Così, la scrittura:

$$D \leftarrow C+A-B;$$

significa: “*sommare i numeri scritti sui foglietti dei cassette C e A, sottrarre alla somma il numero scritto sul foglietto del cassetto B, scrivere il risultato su un nuovo foglietto e inserire questo foglietto nel cassetto D dopo aver eliminato il foglietto (eventualmente) presente in D*”.

N.B. Per brevità diciamo, ad esempio, “*il numero contenuto in C*” invece di “*il numero scritto sul foglietto contenuto in C*”.

2. L’ASPETTO FORMALE

Invece di parlare di cassette e di numeri scritti su foglietti (come nell’esercizio precedente), si può ricorrere a un’altra descrizione, più astratta.

Le lettere maiuscole A, B, C, ... sono chiamate “*variabili*” (invece di cassette) e i numeri sui foglietti sono detti “*valori*” di quelle variabili. La sequenza di calcoli dell’ESERCIZIO precedente può essere presentata come la *procedura* seguente.

```

Procedura PROVA1;
variables A, B, C, D, E, F integer;
input A, B, C;
D ← A+B;
E ← C+B-A;
F ← A+B-C;
output D, E, F;
endprocedura;
```

Con la scrittura “variables A, B, C, D, E, F, integer” si dice che esistono sei cassette (detti appunto variabili) e che sui foglietti in ognuno dei cassette può essere scritto (solo) un numero intero.

Con la scrittura “input” si assegnano dei valori a certe variabili (si scrivono dei numeri sui foglietti contenuti in certi cassette).

Con la scrittura “output” si fa vedere il valore di certe variabili (si leggono i numeri sui foglietti contenuti in certi cassette).

Se si *esegue* la procedura PROVA1 e in input alle variabili A, B, C vengono assegnati rispettivamente i valori 5, 8 e 3, in output, per le variabili D, E, F vengono resi visibili rispettivamente i valori 13, 6 e 10 che sono soluzione del problema precedente, di cui PROVA1 è la trascrizione *formale*.

N.B. Ogni riga della procedura si dice *statement* (o *istruzione*).

3. UN NUOVO ESEMPIO

Si ricordi che quando si assegna un nuovo valore a una variabile (cioè si inserisce un nuovo foglietto in un cassetto) l'eventuale valore in essa preesistente viene distrutto (cioè il vecchio foglietto viene buttato e non è più recuperabile).

Ricapitolando, con le lettere A, B, C, ... (o in generale con nomi scritti con lettere maiuscole e numeri) si indicano, in una *procedura*, delle *variabili* che possono acquisire valori mediante

una *istruzione* (o *statement*) "input",

una *istruzione* (o *statement*) di *assegnazione*.

Si consideri la procedura ESEMPIO seguente, brevemente commentata.

procedura	commento
procedure ESEMPIO1;	inizio della procedura di nome ESEMPIO
variables A, B, C1, D integer;	si dichiara che si usano 4 variabili che assumono valori interi
input A, B;	si acquisiscono dall'esterno valori per le variabili A e B
C1 ← A+B;	la variabile C1 acquisisce valore (di una espressione)
D ← C1+B-A;	la variabile D acquisisce valore (di una espressione)
A ← C1+D;	la variabile A acquisisce (un nuovo) valore (di una espressione)
output A, C1, D;	si rendono disponibile all'esterno i valori delle variabili A, C1, D
endprocedure;	fine della procedura

Se in input alle variabili A e B vengono assegnati rispettivamente i valori 5 e 9, in output vengono restituiti i valori 32, 14 e 18 rispettivamente per A, C1, D.

4. L'ALTERNATIVA "if"

Durante la svolgimento di calcoli in una procedura si può porre una "alternativa" decisa dal valore di un *predicato*: se il predicato è *vero* si fanno alcune cose, se è *falso* se ne fanno altre. In una procedura, l'alternativa può essere descritta, per esempio, con la seguente struttura

variables A, B, M integer;

...

if A > B

 then M ← A;

 else M ← B;

endif;

output M;

...

Se per esempio il valore di A è 2 e quello di B è 5, dopo l'alternativa il valore in *output* di M è 5. Naturalmente al posto di "A > B" si possono usare altri predicati, costruiti confrontando i valori di certe variabili: per esempio "A = B" oppure "A < B".

Quando *manca* il ramo "else" (cioè quando occorre fare alcune cose se il predicato è *vero*, ma non si deve fare nulla se è *falso*), si può usare la forma abbreviata del costrutto "if" come nel seguente esempio (che ha lo stesso significato del precedente):

variables A, B, M integer;

```

...
M ← B;
if A > B then M = A; endif;
output M;
...

```

SECONDA PARTE

5. LA RIPETIZIONE “for”

In una procedura si può prevedere di eseguire un insieme di operazioni (detto ciclo) un certo numero di volte; nell'esempio che segue il ciclo è fatto di due operazioni che vengono ripetute 4 volte:

```

procedura ESEMPIO;
variables A, B, K integer;
A ← 0;
B ← 0;
for K from 1 to 4 step 1 do;
    A ← A+K;
    B ← B+K×K;
endfor;
output A, B;

```

procedura	commento
<pre> procedura ESEMPIO2; variables A, B, K integer; A ← 0; B ← 0; for K from 1 to 4 step 1 do; A ← A+K; B ← B+K×K; endfor; output A, B; </pre>	<p>inizio del ciclo (che è ripetuto 4 <i>volte</i> con i valori di K: 1, 2, 3, 4) primo statement del ciclo (A assume via via i valori 1, 3, 6, 10) secondo statement del ciclo (B assume via via i valori 1, 5, 14, 30) segnala che il ciclo arriva fin qui</p>

I valori di output sono: per A 10 (la somma dei 4 valori di K) e per B 30 (la somma dei quadrati dei valori di K).

6. LA RIPETIZIONE “while”

La ripetizione di un gruppo di azioni può essere comandata non solo con la struttura “for” già vista, ma anche con la struttura “while”, illustrata dal seguente esempio.

```

B ← 10;
A ← 0;
K ← 0;
while A < B do;
    K ← K + 1;
    A ← K × K + A;
endwhile;
output A;

```

Se il predicato $A < B$ è vero, il ciclo viene ripetuto; quando diventa falso si passa alla esecuzione della istruzione successiva a “endwhile”. In questo caso il valore di B rimane fisso a 10, mentre quello di A cambia dopo ogni iterazione assumendo i seguenti valori: 1, 5, 14. Dopo la terza

iterazione il valore di A non è più minore di quello di B e il ciclo si arresta: in output si ha quindi 14.

7. VARIABILI REALI

Oltre a valori interi le variabili possono contenere valori razionali, cioè numeri “con la virgola”: in questo contesto, però, si userà sempre il “.” come separatore decimale. Le variabili di questo tipo si dicono “float”. Corrispondentemente alle variabili di tipo float si usano le costanti di tipo float: si scrivono col punto decimale seguito da almeno una cifra, come in

5.45
5.0
45362.9877

Il seguente è un esempio di procedura che usa variabili float.

procedura	commento
<pre> procedura ESEMPIO3; variables A, B, C integer; variables TF, SF, R float; B ← 2; A ← 6; TF ← 2.0; SF ← 5.0; C ← A/B; R ← SF/TF; output C, R; endprocedura;</pre>	<p>dichiarazione di variabili intere dichiarazione di variabili razionali assegnazione (←) di costante intera assegnazione (←) di costante intera assegnazione (←) di costante razionale assegnazione (←) di costante razionale calcolo e assegnazione (←) con valori interi calcolo e assegnazione (←) con valori razionali per C l’output è 3; per R l’output è 2.5</p>

Si ricorda che le normali operazioni aritmetiche sono indicate con +, −, ×, / (talvolta con ÷). Spesso è usato anche l’elevamento a potenza, col simbolo ^. Per esempio $2^3 = 8$; A^B : se A ha valore 2 e B ha valore 3, il risultato dell’espressione è 8; se A ha valore 2.0 e B ha valore 3, il risultato dell’espressione è 8.0.

7. VARIABILI STRINGA

Oltre a valori numerici le variabili possono contenere *stringhe*, cioè sequenze di caratteri. Le variabili di questo tipo si dicono “string”. Corrispondentemente alle variabili di tipo string si usano le costanti di tipo string: si scrivono come sequenze di caratteri racchiuse tra apici; per esempio:

‘alpha’
‘Giuseppe’
‘1200’
‘ ’
‘ , ’

N.B. La (costante) stringa ‘1200’ *non* è il numero intero 1200: in particolare non si può sommare o sottrarre; la costante ‘ ’ è la stringa *vuota*; la costante ‘ ’ è la stringa *spazio*;
Consideriamo solo tre operazioni tra stringhe: la concatenazione, la estrazione e la determinazione della lunghezza.

I seguenti sono esempi di procedure che usano variabili string e intere.

procedura	commento
<pre> procedura ESEMPIO4; variables L, integer; variables AS, BS, CS, DS string; BS ← 'Aprile'; L ← ? BS; AS ← 'mese di ' BS; CS ← (1,3) BS; DS ← (4,L) BS; endprocedura;</pre>	<p>dichiarazione di variabile intera dichiarazione di variabili stringhe</p> <p>assegnazione (\leftarrow) di costante string assegnazione (\leftarrow) di valore intero e calcolo della lunghezza della stringa; l'operazione ? calcola la lunghezza del valore (stringa) di BS</p> <p>assegnazione (\leftarrow) di valore stringa, ottenuto concatenando una costante e (il valore di) una variabile: AS vale 'mese di Aprile': l'operazione è il concatenamento</p> <p>assegnazione (\leftarrow) di valore string: l'operazione (1,3) estrae dalla stringa che è il valore di BS la sottostringa che va dal carattere 1 (primo) al carattere 3 (terzo); il valore di CS è 'Apr'</p> <p>assegnazione (\leftarrow) di valore string: l'operazione (4,L) estrae dalla stringa che è il valore di BS la sottostringa che va dal carattere 4 (quarto) al carattere finale (L vale la lunghezza della lista); il valore di DS è 'ile'</p>

procedura	commento
<pre> procedura ESEMPIO5; variables L, J, J1 integer; variables AS, BS, string; BS ← 'Aprile'; L ← ? BS; for J from 1 to L step 1 do; AS ← (J,J) BS; output AS; endfor; endprocedura;</pre>	<p>dichiarazione di variabile intera dichiarazione di variabili stringhe</p> <p>assegnazione (\leftarrow) di costante string assegnazione (\leftarrow) di valore intero e calcolo della lunghezza della stringa; l'operazione ? calcola la lunghezza del valore (stringa) di BS che è 6</p> <p>ripetizione L volte</p> <p>assegnazione (\leftarrow) di valore string: l'operazione (J,J1) estrae dalla stringa che è il valore di BS ogni volta un carattere (in successione) AS è di volta in volta: 'A', 'p', 'r', 'i', 'l', 'e'</p>

Per il confronto delle stringhe sono disponibili i predicati: = e \neq .

Seguono esempi di problemi che riguardano lo pseudolinguaggio.

PROBLEMA1

Si consideri la seguente procedura PROVA1.

```

procedura PROVA1;
variables A, B, C, D integer;
input A, B;
C ← A + B;
D ← A × B;
A ← C+B;
```

```

B ← (A+B) ×(A- B);
output C, D, A, B;
endprocedure;

```

I valori in input sono: 4 per A, 2 per B; determinare i valori di output di A, B, C, D e scriverli nella seguente tabella.

A	
B	
C	
D	

SOLUZIONE

A	8
B	60
C	6
D	8

COMMENTI ALLA SOLUZIONE

Il problema si risolve eseguendo passo passo le operazioni indicate dalla procedura.

Si noti che i valori di certe variabili cambiano più volte: per esempio le variabili A e B acquisiscono un primo valore nello *statement* input e successivamente cambiano valore con le ultime due operazioni.

PROBLEMA2

Si consideri la seguente procedura PROVA2.

```

procedure PROVA2;
variables A, B, M, N, K integer;
input A;
M ← 0;
N ← 0;
for K from 1 to 10 step 1 do;
    input B;
    if A > B      then M ← M + A;  endif;
    if A < B      then N ← N + A;  endif;
endfor;
output M, N;
endprocedure;

```

I valori di input per A è 5 e per B sono rispettivamente: 9, 3, 7, 2, 8, 5, 1, 4, 4, 5. Determinare i valori di output.

M	
N	

SOLUZIONE

M	25
N	15

COMMENTI ALLA SOLUZIONE

Basta eseguire, passo a passo, le operazioni indicate: in N va la somma di tante volte il valore di A quante volte questo è più piccolo di (quello di) B (5 volte); in M va la somma di tante volte il valore di A quante volte questo è più piccolo di (quello di) B (3 volte).

PROBLEMA3

Si consideri la seguente procedura (scritta in maniera sintatticamente scorretta: il simbolo X non è definito).

```
procedure P1;
variables A, B, C, D integer;
D ← 0;
input A, B, C;
D ← A + B + C + X;
output D;
endprocedure;
```

Trovare, tra le variabili dichiarate nella procedura, il nome da sostituire a X per ottenere in output 21 per D se i valori in input sono 2 per A, 5 per B e 7 per C.

nome della variabile da sostituire a X	
--	--

SOLUZIONE

nome della variabile da sostituire a X	C
--	---

COMMENTI ALLA SOLUZIONE

Poiché A, B, C valgono rispettivamente 2, 5 e 7, occorre che il risultato di

$$A + B + C + X$$

cioè

$$2 + 5 + 7 + X$$

valga 21; questo richiede che il valore di X sia 7, cioè quello di C.

Una altra maniera di procedere alla soluzione è di esaminare tutti i possibili casi: poiché sono dichiarate quattro variabili: A, B, C, D, allora si possono esaminare i risultati delle quattro possibili sostituzioni:

l'espressione	$A + B + C + A$	vale	$2 + 5 + 7 + 2 = 16$
l'espressione	$A + B + C + B$	vale	$2 + 5 + 7 + 5 = 19$
l'espressione	$A + B + C + C$	vale	$2 + 5 + 7 + 7 = 21$
l'espressione	$A + B + C + D$	vale	$2 + 5 + 7 + 0 = 14$

La soluzione segue immediatamente.

PROBLEMA4

Si consideri la seguente procedura PROVA4.

```
procedure PROVA4;
variables J, L, C integer;
variables A, B string;
input A;
```

```

C ← 0;
L ← |?A;
for J from 1, step 1 to L do;
    B ← |(J,J) A;
    if B = 'a' then C ← C + 1; endif;
endfor;
output C;
endprocedure;

```

I valori di input per A è: '9, a, b, 2, ac, 5, a, 4, 4a, 5b', Determinare il valore di output per C

C	
---	--

SOLUZIONE

C	4
---	---

COMMENTI ALLA SOLUZIONE

Basta eseguire, passo a passo, le operazioni indicate: in C va il numero di volte che la lettera 'a' compare nella stringa data in input. Si noti che

|?A

(il numero di caratteri di A) vale 31: naturalmente si contano anche le virgole e gli spazi.