ALLEGATO A - OPS 2016

STRUTTURA DELLE GARE

Il programma OPS per il 2015-2016 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 10 problemi (tale numero potrà variare da 8 a 12); 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 6 problemi (tale numero potrà variare da 5 a 8); la articolazione di norma 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 2015-2016 sono tratti del seguente insieme:

a) Regole e deduzioni;
b) Percorsi in un grafo;
c) *Knapsack*;
d) Pianificazione
e) Statistica elementare;
f) Relazioni tra elementi di un albero;
g) Flussi in una rete;
h) Crittografia;
i) Programmazione dei movimenti di un robot.
j) Movimenti di pezzi degli scacchi.

Di seguito sono riportati esempi dei problemi ricorrenti. Questi esempi sono di carattere elementare; servono solo di riferimento per:

- mantenere "sintetico" l'enunciato dei problemi assegnati effettivamente in gara,
- fissare gli argomenti e la terminologia usata.

Successivamente sono illustrati 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 solo la scuola secondaria.

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.

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 antecedenti>,<conseguente>)

che indica una regola di nome <sigla> che consente di dedurre <conseguente> conoscendo tutti gli elementi contenuti nella <lista 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.

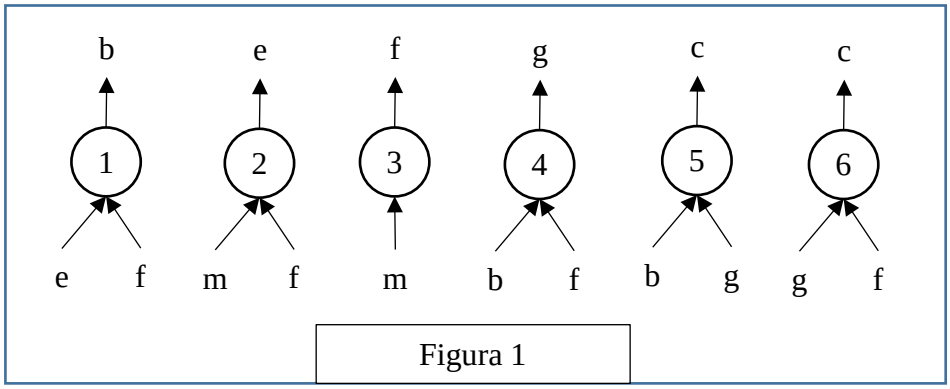


Figura 1

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 "appendere" 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].

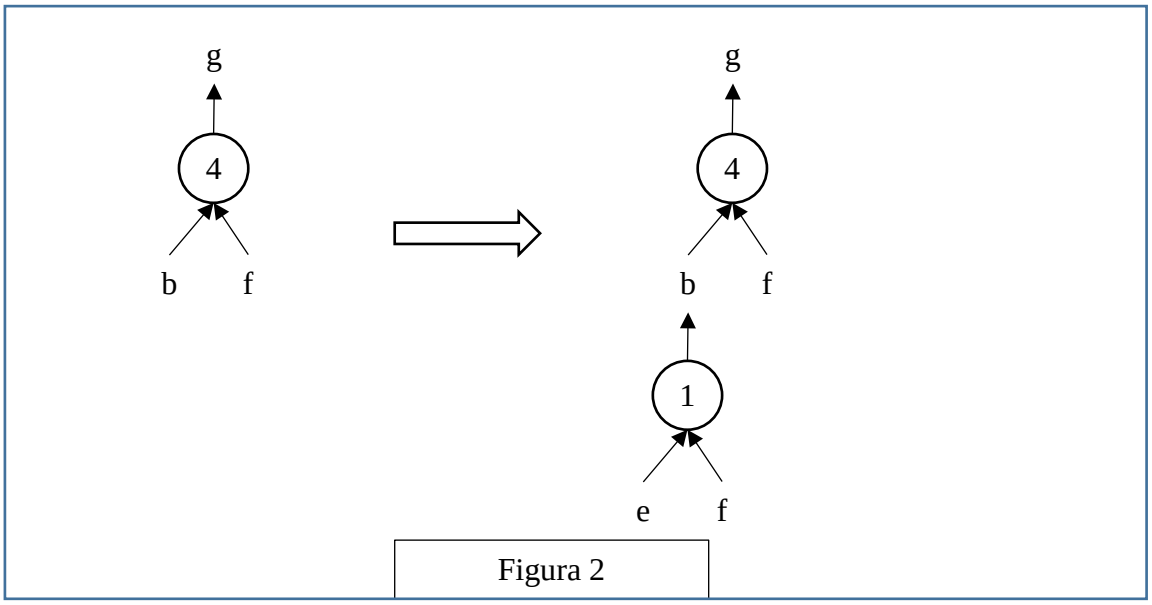


Figura 2

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.

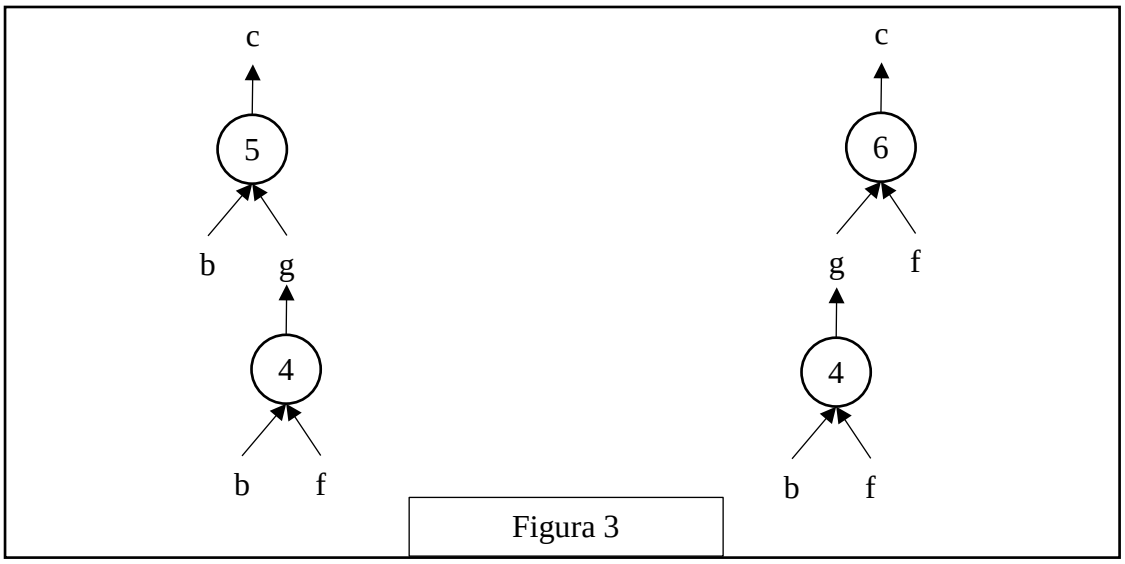


Figura 3

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
Siano date le seguenti regole:

regola(1,[a],h) regola(2,[h,f],b) regola(3,[q,f],h)
regola(4,[a,h],c) regola(5,[d,g],c) regola(6,[b,q],g)
Trovare:

1. la lista L1 che rappresenta il procedimento per dedurre **c** da **a**,
2. la lista L2 che rappresenta il procedimento per dedurre **g** da [**q,f],**

e scriverle nella seguente tabella.

L1	
L2	

SOLUZIONE

L1	[1,4]
L2	[3,2,6]

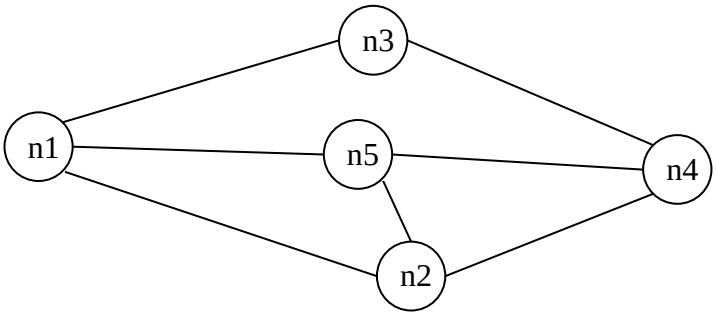
COMMENTI ALLA SOLUZIONE
Nel caso della prima domanda, in cui è dato **a**, si verifica immediatamente che l'unica regola applicabile è la 1, con cui si deduce **h**; con **a** e **h** l'unica regola applicabile è la 4, con cui si deduce **c**, che è la soluzione cercata, quindi la lista L1 è [1,4].
In generale, in casi più complessi, per risolvere questo tipo di problema 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). Nel caso della seconda domanda conviene applicare tale metodo: **g** compare come conseguente solo nella regola 6 che, però, contiene un antecedente, **b**, incognito; questo compare come conseguente solo nella regola 2 che, però, contiene un antecedente, **h**, incognito; questo compare come conseguente solo nella regola 3, che ha come antecedenti (solo) i dati: a questo punto il procedimento si arresta e la lista L2 è [3,2,6].

N.B. Quando si applica il procedimento *backward*, la prima regola che compare nella lista (che rappresenta la soluzione) è l'ultima trovata; la seconda è la penultima e così via.

b) PERCORSI IN UN GRAFO

PREMESSA

Il seguente *grafo* descrive i collegamenti esistenti fra alcune (5) città: queste sono rappresentate da *nodi* di nome n1, n2, …, n5 e i collegamenti sono rappresentati da segmenti, detti *archi*, tra nodi.



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(n1,n2,6)	arco(n1,n3,5)	arco(n3,n4,4)
arco(n1,n5,3)	arco(n2,n4,3)	arco(n2,n5,2)
arco(n5,n4,6)		

Due nodi si dicono *adiacenti* se sono collegati da un arco. 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 [n5,n2,n4,n3] descrive un percorso dal nodo n5 al nodo n3; tale percorso ha lunghezza K = 2 + 3 + 4 = 9.

Un *ciclo* è un percorso che inizia e termina nello stesso nodo, per esempio [n5,n2,n1,n5]. Un percorso si dice *semplice* se *non* ha nodi ripetuti: un percorso semplice, quindi, non contiene cicli; per esempio [n5,n2,n4,n3] è semplice, mentre [n5,n2,n1,n5,n2,n4,n3] non è semplice perché ha nodi ripetuti.

PROBLEMA

È dato un grafo descritto dal seguente elenco di archi:

arco(n1,n2,2)	arco(n2,n3,2)	arco(n3,n1,5)
arco(n4,n1,1)	arco(n4,n2,4)	arco(n4,n5,3)

Disegnare il grafo e trovare:

1. la lista L1 del percorso più breve tra n5 e n3 e calcolarne la lunghezza K1;
2. la lista L2 del percorso più lungo (senza passare più volte per uno stesso nodo) tra n5 e n3 e calcolarne la lunghezza K2.

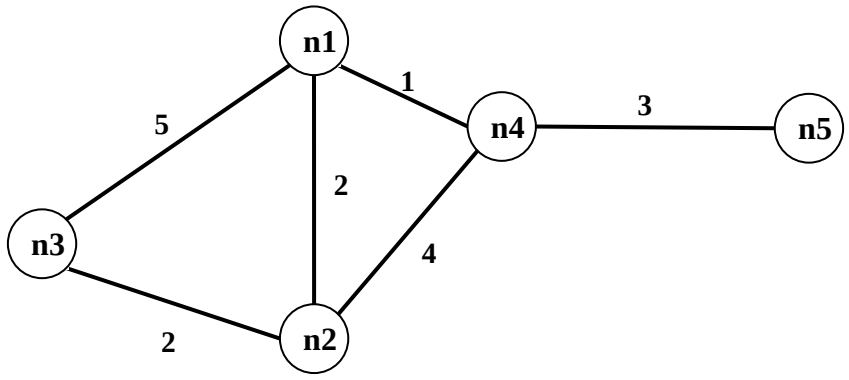
L1	
K1	
L2	
K2	

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.

c) KNAPSACK

PROBLEMA
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< … .

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 la 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.

d) PIANIFICAZIONE

PROBLEMA
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], [A4,A5], [A3,A4], [A5,A6].

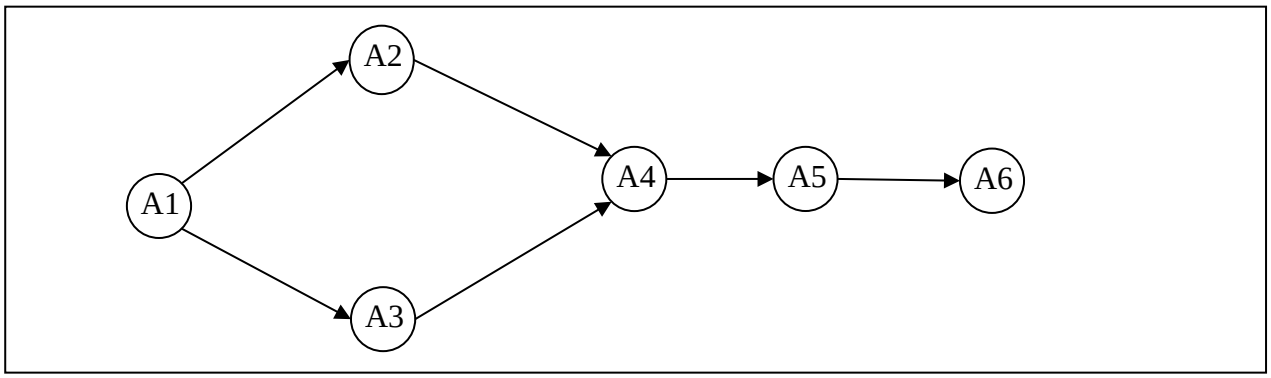
Trovare il numero N di giorni necessari per completare il progetto, tenuto presente che alcune attività possono essere svolte in parallelo e che ogni attività *deve* iniziare prima possibile (nel rispetto delle priorità). Inoltre, trovare il numero massimo RM di ragazzi che lavora contemporaneamente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

N	
RM	

SOLUZIONE

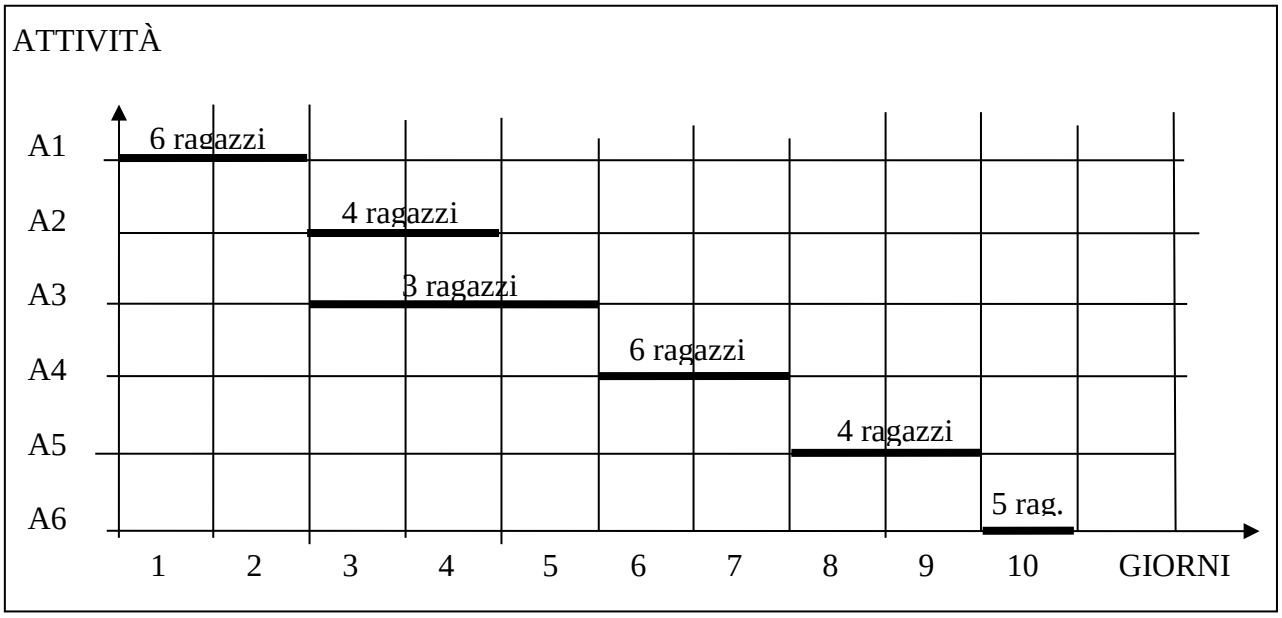
N	10
RM	7

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).

e) STATISTICA ELEMENTARE

PREMESSA

Dato un insieme di numeri, si dice *mediana* il numero che occuperebbe la posizione centrale se l'insieme fosse ordinato. Ad esempio, la mediana dei numeri presenti nella seguente lista [2,3,5,7,6,4,2] è 4; infatti, nella lista ci sono 3 elementi minori di 4 (che sono [2,2,3]) e ce ne sono 3 maggiori di 4 (che sono [5,6,7]).

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

La *media* 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

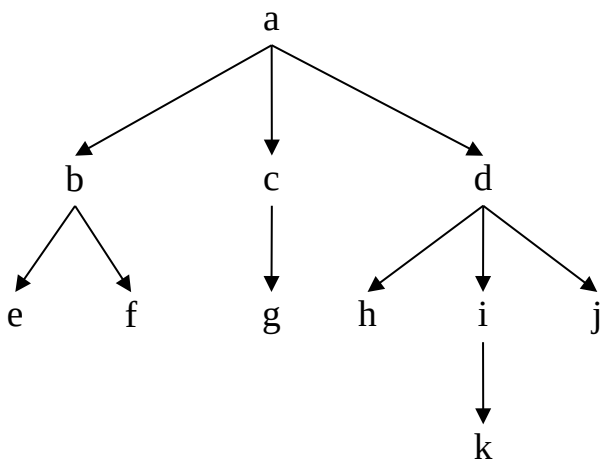
M1	10
M2	7
M3	1

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

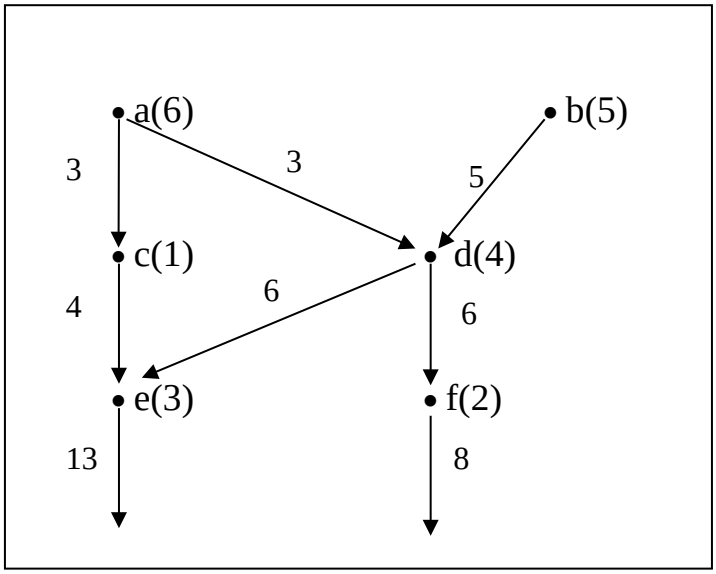
PREMESSA
Sul fianco di una montagna esistono numerose sorgenti. L'acqua di una sorgente, che si suppone fluire in modo continuo e costante, può scorrere a valle attraverso uno o più canali. Può avvenire che uno o più canali convergano in un punto in cui esiste una sorgente; in tal caso, la loro acqua 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(<sorgente>,<litri d'acqua erogata al minuto>),

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

r (<sorgente1>,<sorgente2>),

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 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.

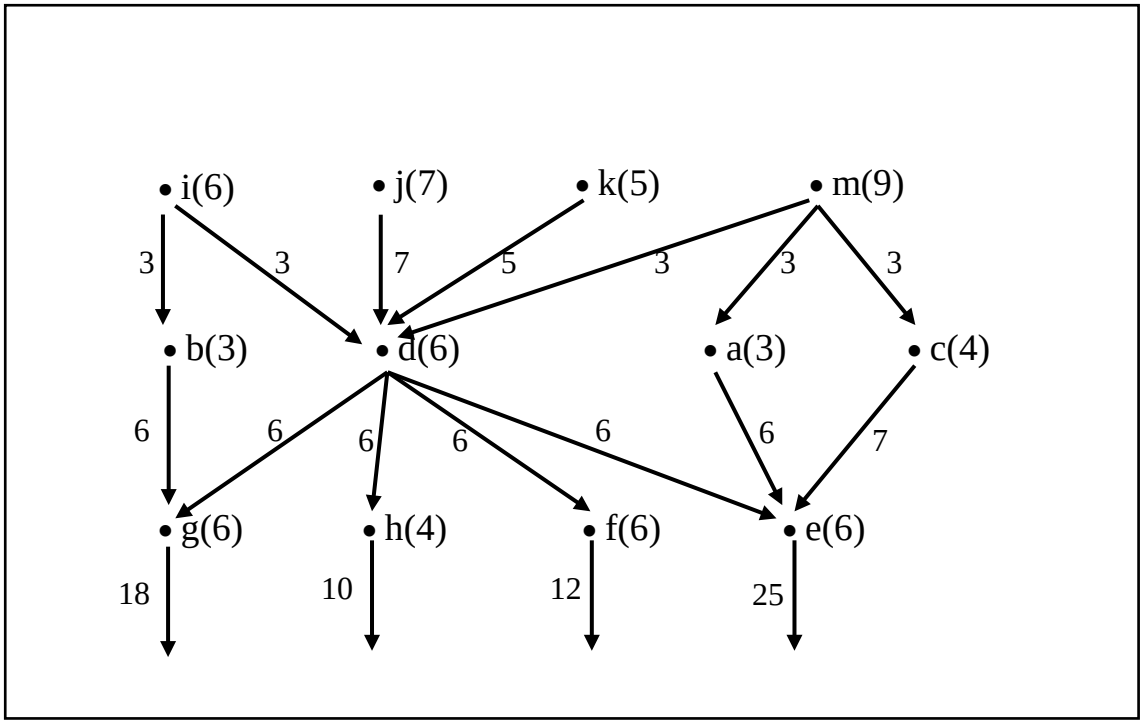
e	
f	
g	
h	

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) PROGRAMMAZIONE DEI MOVIMENTI DI UN ROBOT

PREMESSA
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,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,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 lo stato 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]).

PROBLEMA 1
In un campo di gara, sufficientemente ampio, il robot è nella casella [8,8] con orientamento verso l'alto; deve eseguire il percorso descritto dalla seguente lista di comandi

[f,f,o,f,f,a,f,f,f,o,f].

Trovare l'ascissa X e l'ordinata Y della casella in cui finisce il percorso del robot.

X	
Y	

SOLUZIONE

X	11

Y	13

COMMENTI ALLA SOLUZIONE
La soluzione si costruisce eseguendo uno dopo l'altro i comandi della lista.
Programma: [f,f,o,f,f,a,f,f,f,o,f]

		Posizione e orientamento del robot	
Partenza		[8,8]	verso l'alto
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

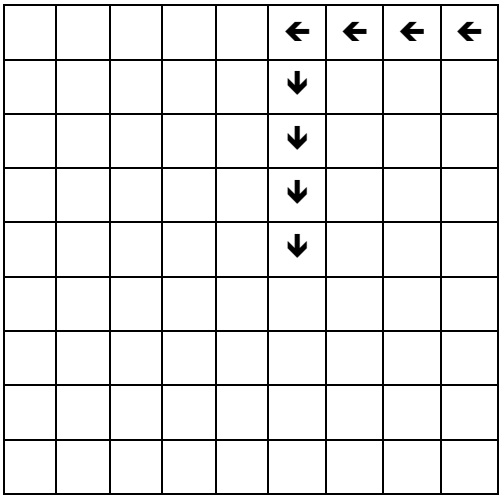
PROBLEMA 2
In un campo di gara il robot è nella casella [9,9] con orientamento verso sinistra: trovare la lista L dei comandi da assegnare al robot per fargli compiere il percorso descritto dalla seguente lista di caselle [[9,9],[8,9],[7,9],[6,9],[6,8],[6,7],[6,6],[6,5]]

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

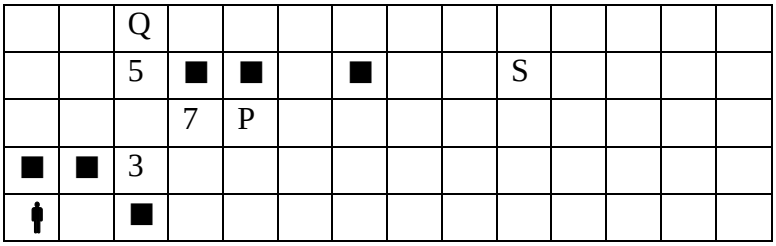
4 a
5 f
6 f
7 f
8 f

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.

j) MOVIMENTO DI PEZZI DEGLI SCACCHI

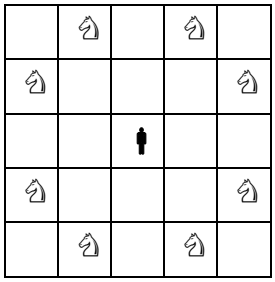
PREMESSA

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



Ogni casella può essere individuata da due numeri (interi); per esempio la casella contenente la lettera P è individuata spostandosi di cinque colonne da sinistra e di tre righe dal basso: brevemente si dice che ha *coordinate* [5,3]; la prima coordinata (in questo caso 5) si dice *ascissa* e la seconda (in questo caso 3) si dice *ordinata*. Le coordinate della casella contenente la lettera S sono [10,4] e di quella contenente il robot ♟ sono [1,1].

Il robot si muove a passi e ad ogni passo (o mossa) può spostarsi solo in una delle caselle contenenti ♘ come illustrato nella seguente figura (allo stesso modo del *cavallo* nel gioco degli scacchi).



Il campo di gara può contenere caselle, segnate da un *quadrato nero* nella prima figura, *interdette* al robot: cioè il robot *non può essere collocato* in quelle caselle (che quindi si comportano come se fossero occupate da un pezzo dello stesso colore del cavallo, nel gioco degli scacchi); quindi, tenuto conto anche dei bordi del campo di gara, la mobilità del robot può essere limitata; ad esempio se il robot si trovasse nella casella in cui c'è Q si potrebbe spostare solo in 3 caselle: non può andare in [5,4] perché è interdetta; se fosse nella casella in cui c'è P avrebbe 7 mosse possibili; dalla casella [1,1] ha solo 2 mosse possibili: in [2,3] e in [3,2].

Un percorso è descritto dalla *lista delle coordinate delle caselle attraversate*; un possibile percorso da P (coordinate [5,3]) a Q (coordinate [3,5]) è descritto dalla lista [[5,3],[3,2],[5,1],[4,3],[3,5]].

In alcune caselle sono posti dei premi che il robot può *raccogliere* lungo un percorso. Ogni premio è descritto fornendo le coordinate della casella che lo contiene e il valore del premio: i premi riportati nella prima figura sono descritti dalla seguente lista [[3,2,3],[4,3,7],[3,4,5]]. Nel percorso da P a Q, sopra descritto, il *totale di premi raccolti* è pari a 10.

PROBLEMA

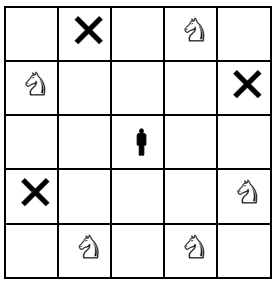
Un campo di gara ha dimensioni 5×5; le caselle interdette descritte dalla seguente lista:

[[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 simbolo ♘) 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].

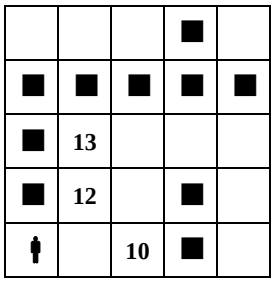
L1	
L2	
N	

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

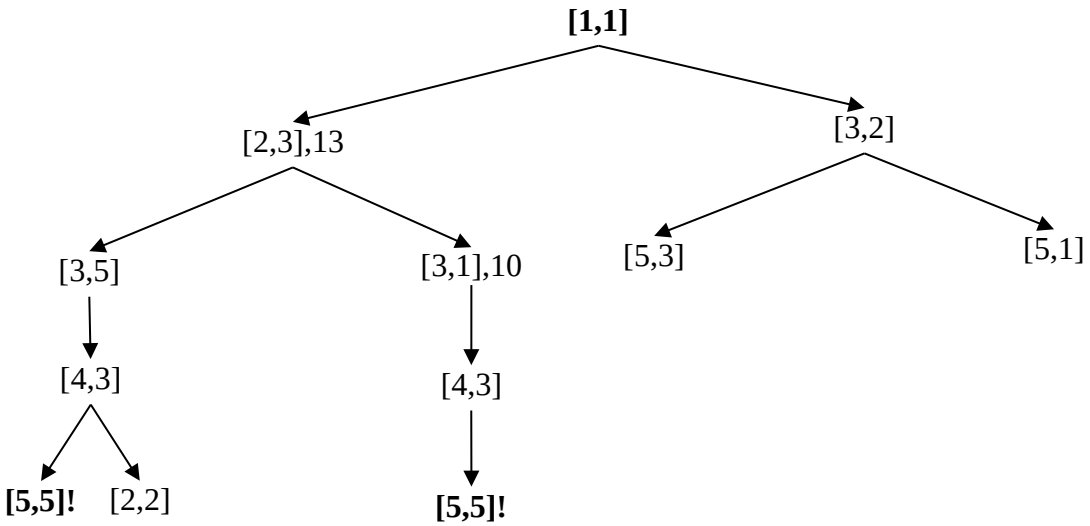
Il campo di gara è mostrato nella figura.



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

ELEMENTI DI PSEUDOLINGUAGGIO

PRIMA PARTE

1. INTRODUZIONE: LA METAFORA

In una cassettiera ci sono dei cassetti 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 ← A+B;

significa: "*sommare i numeri scritti sui foglietti dei cassetti 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 ← C+A-B;

significa: "*sommare i numeri scritti sui foglietti dei cassetti 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 cassetti 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 cassetti) 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.

```
Procedure 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;
endprocedure;
```

Con la scrittura "variables A, B, C, D, E, F, integer" si dice che esistono sei cassetti (detti appunto variabili) e che sui foglietti in ognuno dei cassetti 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 cassetti).

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

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 ESEMPIO;	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 (per le scuole secondarie)

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:

```
procedure 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

procedure 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;
	inizio del ciclo (che è ripetuto 4 *volte* 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

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
procedure ESEMPIO;	
variables A, B, C integer;	dichiarazione di variabili intere
variables TF, SF, R float;	dichiarazione di variabili razionali
B ← 2;	assegnazione (←) di costanti intere
A ← 6;	assegnazione (←) di costanti intere
TF ← 2.0;	assegnazione (←) di costanti razionali
SF ← 5.0;	assegnazione (←) di costanti razionali
C ← A/B;	calcolo e assegnazione (←) con valori interi
R ← SF/TF;	calcolo e assegnazione (←) con valori razionali
output C, R;	per C l’output è 3; per R l’output è 2.5
endprocedure;	

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.