Laboratorio di Algoritmi e Strutture Dati

Docenti: M. Torelli, S. Aguzzoli

Appello del 2 luglio 2008

Progetto "Ingranaggi 2"
Consegna entro il 20 luglio 2008

Il problema

L'obiettivo è quello di costruire dei macchinari composti di piastre rettangolari, ciascuna delle quali contiene delle ruote dentate che girano tutte nella stessa direzione. Poiché il senso di rotazione delle ruote di una piastra determina il movimento delle ruote nelle piastre adiacenti, occorre evitare che si creino macchine in cui le ruote siano bloccate.

Il *piano* di riferimento è l'insieme dei punti a coordinate intere. Una piastra P è identificata da un rettangolo nel piano e da un nome σ, che è una parola finita sull'alfabeto $\{a, \ldots, z, 1 \ldots, 9\}$ iniziante per una lettera minuscola $\{a, \ldots, z\}$. Usiamo la notazione $P(x_1, y_1, x_2, y_2, \sigma)$ per indicare una piastra il cui vertice in basso a sinistra ha coordinate (x_1, y_1), il vertice in alto a destra ha coordinate (x_2, y_2) e il cui nome è σ. Più precisamente, l'insieme dei punti della piastra $P(x_1, y_1, x_2, y_2, \sigma)$ è:

$$\{\, (x, y) \in \mathbb{Z}^2 \mid x_1 \leq x \leq x_2,\ y_1 \leq y \leq y_2 \,\}.$$

Date due piastre P_1 e P_2 diciamo che:

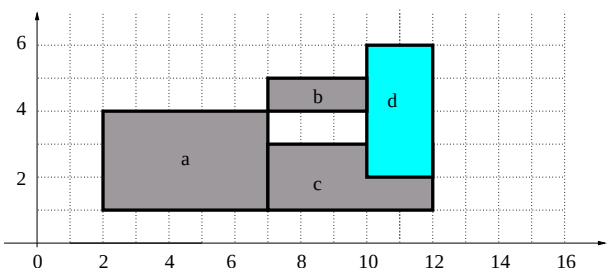
- P_1 e P_2 sono *disgiunte* se $P_1 \cap P_2 = \emptyset$ (non hanno punti in comune);
- P_1 e P_2 sono *adiacenti* se $P_1 \cap P_2 \neq \emptyset$ e tutti i punti in $P_1 \cap P_2$ giacciono su una stessa retta (questo vale in particolare se P_1 e P_2 hanno un solo punto in comune);
- P_1 e P_2 sono *sovrapposte* se $P_1 \cap P_2 \neq \emptyset$ e i punti in $P_1 \cap P_2$ non giacciono su una stessa retta.

Si noti che, date due piastre P_1 e P_2, vale una sola delle proprietà precedenti.

Esempio 1 Consideriamo le piastre

$$P(2, 1, 7, 4, a) \qquad P(7, 4, 10, 5, b) \qquad P(7, 1, 12, 3, c) \qquad P(10, 2, 12, 6, d)$$

rappresentate nella figura qui sotto.



La piastra a è adiacente alle piastre b e c; b è adiacente ad a e d, mentre c è adiacente solamente ad a. Le piastre a e d sono disgiunte, e lo stesso vale per b e c. Le piastre d e c sono sovrapposte.

Un *cammino* dalla piastra P_1 a P_n è una sequenza di piastre $P_1, \ldots, P_n$ tali che per ogni $1 \leq i \leq n-1$ la piastra P_i è adiacente a P_{i+1}, o P_i e P_{i+1} sono una coppia di piastre sovrapposte. Diciamo che un insieme $\mathcal{P}$ di piastre è *connesso* se, per ogni $P, Q \in \mathcal{P}$, esiste un cammino da P a Q contenuto in $\mathcal{P}$. Un *macchinario* è un insieme di piastre $\mathcal{P}$ tale che $\mathcal{P}$ è connesso e, per ogni piastra $Q \notin \mathcal{P}$, $\mathcal{P} \cup \{Q\}$ non è connesso.

Un *cammino di sovrapposizione* dalla piastra P_1 a P_n è una sequenza di piastre $P_1, \ldots, P_n$ tali che per ogni $1 \leq i \leq n-1$ le piastre P_i e P_{i+1} sono sovrapposte. Diciamo che un insieme $\mathcal{P}$ di piastre è *connesso per sovrapposizioni* se, per ogni $P, Q \in \mathcal{P}$, esiste un cammino di sovrapposizione da P a Q contenuto in $\mathcal{P}$. Un *gruppo* è un insieme di piastre $\mathcal{P}$ tale che $\mathcal{P}$ è connesso per sovrapposizione e, per ogni piastra $Q \notin \mathcal{P}$, $\mathcal{P} \cup \{Q\}$ non è connesso per sovrapposizione.

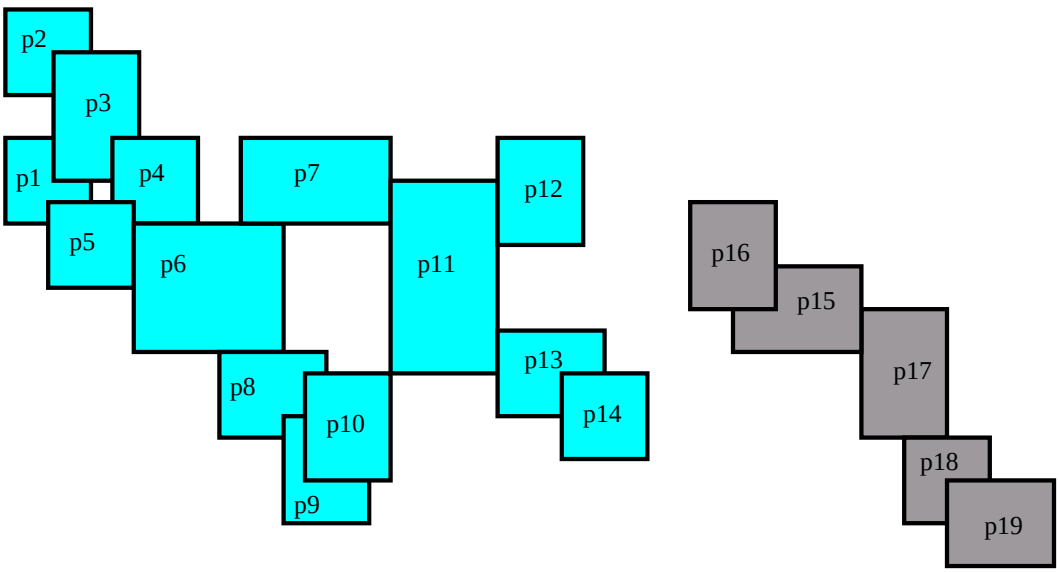
Inizialmente il piano è vuoto. Nel momento in cui una piastra P viene inserita nel piano essa assume una *altezza* $h(P)$ definita in questo modo:

1. Se la piastra P non va a sovrapporsi ad alcun'altra piastra già presente nel piano, allora la sua altezza è $h(P) = 0$.
2. Se la piastra P si sovrappone alle piastre $P_1, P_2, \ldots, P_k$ (già presenti nel piano) e a nessun'altra piastra, allora $h(P) = 1 + \max\{h(P_1), h(P_2), \ldots, h(P_k)\}$.

L'altezza di una piastra non varia durante l'esecuzione del programma. Nell'esempio precedente $h(a) = h(b) = 0$, inoltre, se c è inserita prima di d allora $h(c) = 0$ e $h(d) = 1$; se invece d è inserita prima di c allora $h(d) = 0$ e $h(c) = 1$. In ogni caso vi sono 3 gruppi: $\{a\}$, $\{b\}$ e $\{c, d\}$.

Esempio 2

Supponiamo che, a partire dal piano vuoto, vengano inserite le piastre $p1, p2, \ldots, p19$ come nella figura (le piastre $p4$ e $p6$ sono adiacenti e non sovrapposte)



Allora:

- le piastre aventi altezza 0 sono $p1$, $p2$, $p6$, $p7$, $p8$, $p11$, $p12$, $p13$, $p15$, $p17$, $p18$;
- le piastre aventi altezza 1 sono $p3$, $p9$, $p14$, $p16$, $p19$;
- le piastre aventi altezza 2 sono $p4$, $p10$;
- l'unica piastra con altezza 3 è $p5$.

I gruppi formati da più di una piastra sono:

$$\{p1, p2, p3, p4, p5\} \qquad \{p8, p9, p10\} \qquad \{p13, p14\} \qquad \{p15, p16\} \qquad \{p18, p19\}$$

Abbiamo due macchinari: il macchinario M_1 formato dalle piastre $p1, \ldots, p14$ e il macchinario M_2 formato dalle piastre $p15, \ldots, p19$.

Una piastra P è formata da ruote che possono essere o tutte ferme oppure girare tutte nello stesso senso. Diciamo che P è *ferma* se le sue ruote sono ferme, P *lavora in senso orario* se le sue ruote girano in senso orario, P *lavora in senso antiorario* se le sue ruote girano in senso antiorario. Devono valere le seguenti proprietà:

1. Se P è ferma, tutte le piastre del macchinario di P sono ferme.
2. Se P lavora, tutte le piastre del gruppo di P lavorano nello stesso senso.
3. Se P lavora in senso orario, tutte le piastre adiacenti a P, ma non appartenenti al gruppo di P, lavorano in senso antiorario.
4. Se P lavora in senso antiorario, tutte le piastre adiacenti a P, ma non appartenenti al gruppo di P, lavorano in senso orario.

Se in un macchinario M le proprietà 1, 2 e 3 non possono essere soddisfatte, diciamo che M è *bloccato*. Per evitare situazioni incoerenti, introduciamo due assunzioni sull'inserimento di una piastra.

(A1) Non è possibile inserire una piastra P nel piano se questa diventa adiacente o va a sovrapporsi a una piastra che lavora.

(A2) Non è possibile inserire una piastra P se, dopo il suo inserimento, il macchinario a cui P appartiene è bloccato.

Esempio 3

Consideriamo i macchinari M_1 e M_2 dell'Esempio 2. Se $p1$ lavora in senso orario, allora tutte le piastre di M_1, ad eccezione di $p6$ e $p11$, lavorano in senso orario; le piastre $p6$ e $p11$ lavorano invece in senso antiorario. Se $p15$ lavora in senso antiorario, allora anche $p16$, $p18$ e $p19$ lavorano in senso antiorario, mentre $p17$ lavora in senso orario. Se fermiamo $p7$, tutte le piastre in M_1 si fermano, se fermiamo $p15$, tutte le piastre in M_2 si fermano.

Supponiamo ora che tutte le piastre siano ferme. Non è possibile inserire una nuova piastra p che sia adiacente a $p5$ e $p6$. Infatti, se $p5$ lavora in senso orario, allora $p6$ deve lavorare in senso antiorario, quindi p dovrebbe lavorare sia in senso antiorario (essendo adiacente a $p5$) che orario (essendo adiacente a $p6$), quindi M_1 si bloccherebbe.
Si può invece inserire una piastra q sovrapposta a $p5$ e a $p6$ e disgiunta da tutte le altre piastre del piano. In tal caso viene creato un nuovo gruppo $G = \{p1, p2, p3, p4, p5, p6, q\}$ (si noti che q ha altezza 4). Se ora $p1$ lavora in senso orario, lavorano in senso orario tutte le piastre del gruppo G e la piastra $p11$, le

rimanenti piastre di M_1 lavorano in senso antiorario.
Supponiamo ora di essere nella configurazione iniziale della figura precedente e che tutte le piastre siano ferme. È possibile inserire una piastra r adiacente a $p12$ e $p16$, e disgiunta da tutte le altre piastre del piano. Dopo l'inserimento, viene creato un unico macchinario; se $p1$ lavora in senso orario, le piastre che lavorano in senso antiorario sono $p6$, $p11$, r, $p17$.

Si richiede di implementare una struttura dati efficiente che permetta di eseguire le operazioni seguenti (si tenga presente che la minima porzione rettangolare di piano contenente tutte le piastre può essere molto grande rispetto al numero di piastre presenti nel piano, quindi *non è sicuramente efficiente rappresentare l'insieme delle piastre mediante un'*unica *matrice*).

- **piastra** $(x_1, y_1, x_2, y_2, \sigma)$

  Se esiste già una piastra di nome σ, non esegue alcuna operazione. Altrimenti, inserisce se possibile la piastra $P(x_1, y_1, x_2, y_2, \sigma)$, ossia se se le assunzioni (A1) e (A2) valgono. Se non è possibile effettuare l'inserimento, viene stampato il messaggio

  `errore:` τ `non e' ferma`

  se σ è adiacente o sovrapposta a τ e τ non è ferma

  `errore:` σ `blocca un macchinario`

  se l'aggiunta di σ blocca un macchinario. Nel caso occorrano più violazioni di (A1) o (A2), basta segnalarne una.

  La piastra σ è ferma al momento dell'inserimento.

- **elimina** (σ)

  Se esiste una piastra P di nome σ, elimina P. Altrimenti, non esegue alcuna operazione.

- **elimina** (h)

  Elimina tutte le piastre di altezza h.

- **attiva** (σ, dir)

  Se esiste una piastra P di nome σ, allora P viene attivata in modo che lavori in senso orario se $dir = o$, in senso antiorario se $dir = a$. Altrimenti, non esegue alcuna operazione.

- **disattiva** (σ)

  Se esiste una piastra P di nome σ, allora P e tutte le piastre del macchinario cui P appartiene vengono fermate. Altrimenti, non esegue alcuna operazione.

- **stato** (σ)

  Se esiste una piastra P di nome σ, supponendo che l'altezza di P sia h, stampa uno dei seguenti messaggi a seconda dello stato di P:

  σ h `ferma`
  σ h `lavora in senso orario`
  σ h `lavora in senso antiorario`

  Altrimenti, non esegue alcuna operazione.

- **macchinario** (σ)

  Se esiste una piastra P di nome σ, stampa il nome delle piastre che compongono il macchinario a cui P appartiene secondo il formato specificato nell'apposita sezione. Altrimenti, non esegue alcuna operazione.

All'inizio del programma il piano è vuoto. Si noti che le operazioni richieste sono liberamente implementabili; in particolare, non vanno necessariamente intese come prototipi di funzioni.

Specifiche di implementazione

Il programma deve leggere dallo standard input (`stdin`) una sequenza di righe (separate da `\n`), ciascuna delle quali corrisponde a una riga della prima colonna della Tabella 1, dove σ è una stringa finita sull'alfabeto $\{a, b, \ldots, z, 1, \ldots, 9\}$ di lunghezza *arbitraria* il cui primo carattere appartiene a $\{a, b, \ldots, z\}$; x_1, x_2, y_1, y_2 sono interi; h è un intero positivo e d è il carattere `o` o il carattere `a`. I vari elementi sulla riga sono separati da uno o più spazi. Quando una riga è letta, viene eseguita l'operazione associata; le operazioni di stampa sono effettuate sullo standard output (`stdout`), e ogni operazione deve iniziare su una nuova riga.

Riga di input	Operazione
`p` x_1 y_1 x_2 y_2 σ	**piastra** $(x_1, y_1, x_2, y_2, \sigma)$
`e` σ	**elimina** (σ)
`e` h	**elimina** (h)
`a` σ d	**attiva** (σ, d)
`d` σ	**disattiva** (σ)
`s` σ	**stato** (σ)
`m` σ	**macchinario** (σ)
`f`	Termina l'esecuzione del programma

Tabella 1: Specifiche del programma

Note

1. Non devono essere presenti vincoli sul numero di piastre e macchinari, sulla loro collocazione nel piano, sulla loro dimensione e altezza, sulla lunghezza dei nomi di piastre (se non quelli determinati dal tipo di dato intero). Non si richiede – anzi si sconsiglia – l'uso di grafica, se non per test personali: in modo particolare, non si usi `conio.h` e neppure `clrscr()`.

2. Per semplicità si suppone che l'input sia sempre conforme alle specifiche di Tabella 1, per cui non è necessario controllare la correttezza dell'input. Per leggere l'input si usino le funzioni standard ANSI C `getchar()` e/o `scanf()`.

3. Si consideri il comando
   `m` σ
   e siano $\sigma_1, \sigma_2, \ldots, \sigma_k$ le piastre che compongono il macchinario cui appartiene la piastra σ, elencate in un ordine arbitrariamente scelto, e supponiamo che le rispettive altezze siano $h_1, h_2, \ldots, h_k$. Allora l'output del comando deve essere:

(
σ_1 h_1
σ_2 h_2
$\vdots$
σ_k h_k
)

Esempio
Si supponga che le righe di input siano:

```
p -10 4 -4 8 alfa
p -16 -5 -13 -2 beta
p -5 7 -3 9 gamma
p -13 -2 -10 4 delta
a gamma o
s beta
m beta
p -10 -5 -9 2 epsilon
d delta
p -10 -5 -9 2 epsilon
p -13 -8 -7 -5 zeta
a gamma a
s epsilon
s zeta
m zeta
d gamma
p -8 -10 -6 6 eta
e delta
m zeta
p -8 -10 -6 6 eta
a gamma a
s beta
d gamma
p -13 -2 -10 3 delta
a gamma a
s delta
d gamma
p -11 2 -1 4 theta
a gamma a
s beta
m gamma
d beta
p -4 0 1 10 iota
s iota
e 1
e 2
m iota
p -5 7 -3 9 gamma
m iota
```

```
a gamma o
s alfa
s delta
p -12 -5 -11 -2 kappa
p -14 -4 -9 -3 lambda
s lambda
a gamma a
s lambda
a beta a
s lambda
m lambda
d gamma
d lambda
p -11 -10 2 2 mu
a gamma a
s kappa
m kappa
f
```

L'output prodotto dal programma deve essere il seguente

```
beta 0 lavora in senso orario
(
alfa 0
beta 0
gamma 1
delta 0
)
errore: delta non e' ferma
epsilon 0 lavora in senso antiorario
zeta 0 lavora in senso orario
(
alfa 0
beta 0
gamma 1
delta 0
epsilon 0
zeta 0
)
errore: eta blocca un macchinario
(
beta 0
epsilon 0
zeta 0
)
beta 0 lavora in senso orario
delta 0 lavora in senso antiorario
beta 0 lavora in senso orario
(
alfa 0
```

```
beta 0
gamma 1
delta 0
epsilon 0
zeta 0
eta 1
theta 2
)
iota 3 ferma
(
alfa 0
iota 3
)
(
alfa 0
gamma 4
iota 3
)
alfa 0 lavora in senso orario
delta 0 ferma
lambda 1 ferma
lambda 1 ferma
lambda 1 lavora in senso antiorario
(
beta 0
delta 0
epsilon 0
zeta 0
kappa 0
lambda 1
)
kappa 0 lavora in senso antiorario
(
alfa 0
beta 0
gamma 4
delta 0
epsilon 0
zeta 0
iota 3
kappa 0
lambda 1
mu 4
)
```

Presentazione del progetto

Il progetto deve essere inviato per posta elettronica all'indirizzo `aguzzoli@dsi.unimi.it` entro il 20 luglio 2008 (incluso). La discussione del progetto e l'esame orale si svolgeranno in data e luogo da specificarsi

(consultare al riguardo il sito: `http://homes.dsi.unimi.it/~aguzzoli/algo.htm`).

Occorre presentare:

1. il codice sorgente (rigorosamente ANSI C, compilabile con **gcc**);
2. una sintetica relazione (formato pdf o rtf) che illustra le strutture dati utilizzate e analizza il costo delle diverse operazioni richieste dalla specifica.

I due o più file (file sorgenti C + relazione) devono essere contenuti in un unico file `.zip` il cui nome dovrà essere `cognome.zip`. La relazione e il codice devono riportare il vostro nome, cognome e matricola.

Una copia cartacea della relazione e del codice deve inoltre essere consegnata al dr. Aguzzoli entro il 20 luglio 2008 (lasciandola eventualmente nella sua casella postale presso il dipartimento in via Comelico).

Si ricorda infine di presentarsi alla prova orale con una copia stampata della relazione e del codice.

La discussione del progetto e l'esame orale di Algoritmi e Strutture Dati si svolgeranno indicativamente nei giorni 23, 25 e 28 luglio 2008.

Alla consegna del progetto, indicare nel testo della e-mail la data in cui si preferisce sostenere la prova orale; nei limiti del possibile si cercherà di tener conto di tali indicazioni (se non si hanno preferenze, non dare alcuna indicazione).

Il calendario degli esami orali sarà disponibile sulla pagina del corso qualche giorno dopo il termine di consegna del progetto.

Per ogni ulteriore chiarimento:
E-mail: `aguzzoli@dsi.unimi.it`
Ricevimento: il mercoledì, ore 15-16, stanza S204.

Avvisi

La versione aggiornata del progetto è pubblicata in .pdf sul sito:
`http://homes.dsi.unimi.it/~aguzzoli/algo.htm`.
Si consiglia di consultare periodicamente questo sito per eventuali correzioni e/o precisazioni relative al testo del progetto.

Si richiede allo studente di effettuare un adeguato collaudo del proprio progetto su numerosi esempi diversi per verificarne la correttezza e valutarne le prestazioni.

La realizzazione del progetto è una prova d'esame da svolgersi **individualmente**. I progetti giudicati frutto di **collaborazioni** saranno **estromessi** d'ufficio dalla valutazione.