

# Algoritmi e programmazione avanzata

## Compito di teoria

giovedì 23 gennaio 2003

Tempo: 45 minuti, NON è possibile consultare libri o appunti.

### Esercizio 1 (3 punti)

Sia data la sequenza di interi supposta memorizzata in un vettore:

12 23 19 70 42 51 99 24 8 92

La si trasformi in un heap, ipotizzando di usare un vettore come struttura dati. Si riportino graficamente i diversi passi della costruzione dell'heap ed il risultato finale. Si ipotizzi che, alla fine, nella radice dell'heap sia memorizzato il valore minimo.

### Esercizio 2 (3 punti)

Si inseriscano in sequenza le chiavi

12 71 11 32 45 50 61 12 39 38

in un BST supposto inizialmente vuoto.

### Esercizio 3 (3 punti)

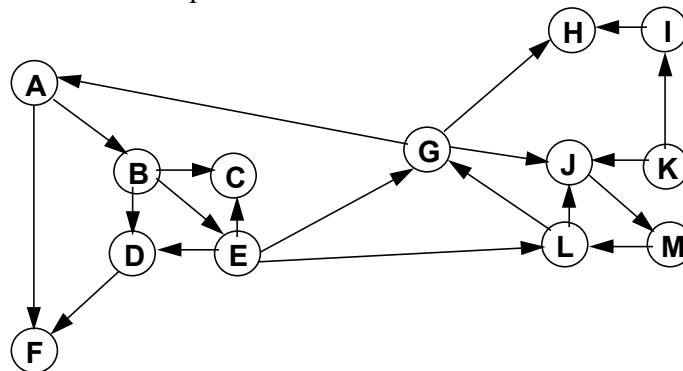
Si consideri una tabella di hash basata su *open addressing* che utilizza la tecnica del *linear probing* per risolvere le collisioni. La tabella è inizialmente vuota. Su di essa viene eseguita la seguente sequenza di operazioni di inserimento di elementi con chiave corrispondente ad una stringa:

EOMER EOWYN THEODEN FRED A MADRIL GRISHNAKH HALETH

Si mostri il contenuto della tabella di hash al termine della sequenza di inserimenti, assumendo che la sua dimensione sia pari a  $M=9$  e che la funzione di hash utilizzata corrisponda per ciascuna chiave alla somma modulo  $M$  dei numeri d'ordine dei caratteri che compongono la stringa all'interno dell'alfabeto italiano (A=0, B=1, C=2, ecc.).

### Esercizio 4 (6 punti)

Sia dato il seguente grafo orientato e pesato.



#### Punto a (2 punti)

se ne effettui una visita in profondità, ritornando come risultato l'albero della visita in profondità

#### Punto b (2 punti)

se ne effettui una visita in ampiezza, ritornando come risultato l'albero della visita in ampiezza.

#### Punto c (2 punti)

Si determini un minimum spanning tree del seguente grafo non orientato pesato, ritornando come risultato l'albero e il valore del peso minimo applicando l'algoritmo di Kruskal.

Qualora necessario, si trattino i vertici secondo l'ordine alfabetico.

# Algoritmi e programmazione avanzata

## Compito di programmazione

giovedì 23 gennaio 2003

Tempo: 60 minuti, è possibile consultare libri e appunti.

Un file di testo ha righe di lunghezza uguale a 80 caratteri e un numero di righe indefinito (e arbitrariamente elevato). Esso contiene solo tre tipi di caratteri: il punto '.', l'asterisco '\*', e il carattere di "a capo".

Occorre scrivere un programma in linguaggio C che rintracci il più grande insieme di asterischi adiacenti. Dato un asterisco, un secondo asterisco è ad esso adiacente se si trova in una delle 8 posizioni contigue (stessa riga subito a sinistra o a destra, stessa colonna in alto o in basso, lungo la diagonale in alto o in basso, lungo la diagonale inversa in alto o in basso).

Al termine del procedimento occorre visualizzare la dimensione più grande insieme (vedere l'esempio per ulteriori chiarimenti).

Il nome del file di ingresso va letto da tastiera.

Esempio

Sia il file il seguente (con righe costituite da 20 caratteri per semplicità):

```
.. ** ..... * .....
.. *** ..... * .....
... **** ..... * .....
..... * .....
..... ** .....
..... **** .....
.. ** ..... ***** .....
.. ** ..... * .....
..... * .....
..... ***** .....
.....
..... *
```

Il più grande insieme è di dimensione 12 (in grassetto nell'esempio)