

Algoritmi e Programmazione Avanzata

Compito d'esame del 25-9-2002

Parte I (Teoria)

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

1 (3 punti)

Si inserisca la seguente sequenza di interi in una coda a priorità:

21 17 51 32 45 55 61 29 39 38 37 33 19 81 18

Si ipotizzi di usare uno heap come struttura dati. Si disegni la struttura dello heap al termine. Si ipotizzi che la priorità massima sia associata alla chiave con valore minimo.

2 (3 punti)

Si inseriscano in sequenza le chiavi 21 17 51 32 45 55 61 29 39 38 in un BST supposto inizialmente vuoto.

3 (3 punti)

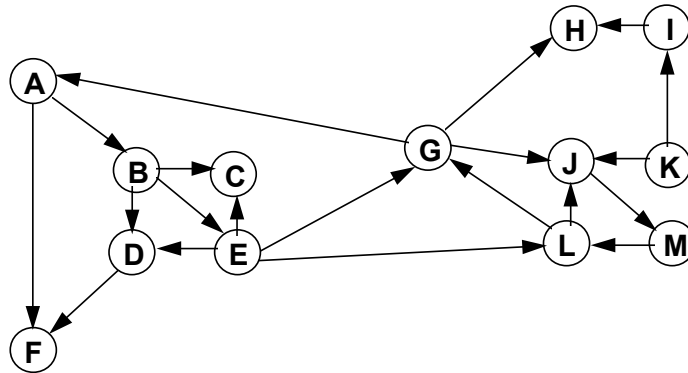
Sia data la sequenza di chiavi SECONDOAPPELLO, dove ciascun carattere rappresenta una chiave ed è individuato dal suo ordine progressivo nell'alfabeto (A=1, ..., Z=26). Si riporti la struttura di una tabella di hash di dimensione 23, inizialmente supposta vuota, in cui avvenga l'inserimento della sequenza di cui sopra. Si supponga di utilizzare per la risoluzione delle collisioni il metodo dell'open addressing con linear probing.

4

Sia dato il seguente grafo orientato in figura. Considerando G come vertice di partenza:

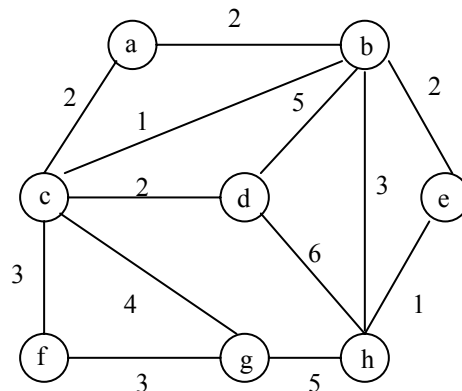
- se ne effettui una visita in profondità, ritornando come risultato l'albero della visita in profondità (2 punti)
- se ne effettui una visita in ampiezza, ritornando come risultato l'albero della visita in ampiezza (2 punti).

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



5 (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.



Algoritmi e Programmazione Avanzata

Compito d'esame del 25-9-2002

Parte II (programmazione)

Tempo: 120 minuti, è possibile consultare libri o 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 la più lunga sequenza 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 della sequenza più lunga (vedere l'esempio per ulteriori chiarimenti).

Si osservi inoltre che:

- le sequenze non si biforcano
- ciascun asterisco ha al più due asterischi adiacenti
- sequenze diverse non si sovrappongono.

È consentito memorizzare il file in una matrice allocata staticamente.
Il nome del file di ingresso va letto da tastiera.

Esempio

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

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

La sequenze di asterischi di lunghezza massima è lunga 9.