

Algoritmi e programmazione avanzata

Compito di teoria

2 Marzo 2002

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

Esercizio 1

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

50 70 37 45 20 30 18 22 36 65

Punto a (3 punti)

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.

Punto b (2 punti)

Si eseguano su tale heap i primi 2 passi dell'algoritmo di heapsort, riportando graficamente i passi dell'algoritmo.

Esercizio 2

Sia data la sequenza di chiavi MAISTRAC, dove ciascun carattere è individuato dal suo ordine progressivo nell'alfabeto (A=1, ..., Z=26). Si riporti la struttura di una tabella di hash di dimensione 19, inizialmente supposta vuota, in cui avvenga l'inserimento della sequenza di cui sopra.

Punto a (1 punto)

Si definisca la funzione di hash che meglio si adatta al problema.

Punto b (1 punto)

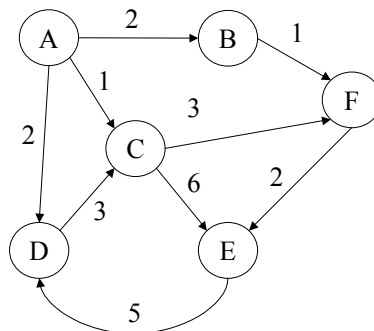
Si esegua l'inserimento delle chiavi di cui sopra supponendo di utilizzare il chaining per la gestione delle collisioni.

Punto c (1 punto)

Si esegua l'inserimento delle chiavi di cui sopra supponendo di utilizzare l'open addressing con probing lineare.

Esercizio 3

Sia dato il seguente grafo orientato e pesato.



Punto a (1 punti)

Si disegnano la matrice di adiacenze e la lista di adiacenze del grafo.

Punto b (2 punti)

Si visiti il grafo in *profondità* ed in *ampiezza*, considerando A come vertice di partenza. Qualora necessario, si trattino i vertici secondo l'ordine alfabetico.

Punto c (4 punti)

Si determinino i valori di tutti i cammini minimi che collegano il vertice A con ogni altro vertice mediante l'algoritmo di Dijkstra. Si assuma, qualora necessario, un ordine alfabetico per i vertici e gli archi.

Algoritmi e programmazione avanzata

Compito di programmazione

19 Marzo 2002

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

La clinica 7 chili in 7 giorni ha deciso di computerizzare la gestione delle diete dei suoi pazienti. In particolare, nella cartella clinica di un paziente, memorizzata in un file, è memorizzato l'elenco delle pietanze che il paziente può mangiare e per le quali non presenta reazioni allergiche. Il file con la cartella clinica del paziente ha il seguente formato:

```
Numero pietanze ammissibili
Codice pietanza numero 1
Codice pietanza numero 2
.
.
```

In un secondo file è memorizzato l'elenco delle pietanze che la mensa della clinica può cucinare, memorizzato nel seguente formato:

```
Numero pietanze
codice pietanza numero 1      numero calorie
codice pietanza numero 2      numero calorie
.
.
```

Si assuma che il codice delle pietanze ed il numero di calorie siano numeri interi.

Opzione a (max 12 punti)

Supponendo che ad ogni pasto il paziente debba mangiare esattamente M pietanze, si scriva una procedura in linguaggio C in grado di identificare un elenco di M pietanze compatibili con il paziente e il numero totale di calorie assunte nel pasto.

Opzione b (max 15 punti)

Supponendo che ad ogni pasto il paziente debba mangiare esattamente M pietanze, si scriva una procedura in linguaggio C in grado di calcolare l'elenco delle M pietanze, compatibili con il paziente, che permetta di minimizzare il numero di calorie assunte durante il pasto.

Esempio

Cartella clinica del paziente:

```
3 /* Numero pietanze ammissibili */
1 /* Codice pietanza numero 1 */
3 /* Codice pietanza numero 3 */
5 /* Codice pietanza numero 5 */
```

Elenco pietanze disponibili:

```
5 /* Numero pietanze */
1 2000 /* codice pietanza numero 1 numero calorie */
2 1000 /* codice pietanza numero 2 numero calorie */
3 200 /* codice pietanza numero 3 numero calorie */
4 400 /* codice pietanza numero 4 numero calorie */
5 600 /* codice pietanza numero 5 numero calorie */
```

Supponendo $M=2$, si ha:

Opzione a Menu: 1, 3 Totale calorie: 2200	Opzione b Menu: 3, 5 Totale calorie: 800
--	---