

# 03MNO Algoritmi e Programmazione

Appello del 26/06/2017 - Prova di teoria (12 punti)

## 1. (1 punto)

Sia data la seguente sequenza di coppie, dove la relazione  $i-j$  indica che il nodo  $i$  è adiacente al nodo  $j$ :  
0-5, 1-3, 0-2, 4-0, 7-6, 3-4, 9-8, 1-3, 6-7, 10-2

si applichi un algoritmo di on-line connectivity con quickfind, riportando a ogni passo il contenuto del vettore e la foresta di alberi al passo finale. I nodi sono denominati con interi tra 0 e 10.

## 2. (1 punto)

Si ordini in maniera ascendente mediante merge-sort il seguente vettore di interi:

31 19 2 14 43 3 89 23 29 17 51 10 18 16 8 100

Si indichino i passaggi rilevanti.

## 3. (2 punti)

Sia data una coda a priorità inizialmente vuota implementata mediante uno heap. Sia data la sequenza di interi e carattere \*: 20 12 9 17 \* \* 85 \* 71 29 31 \* \* 14 \* 41 27 38 dove ad ogni intero corrisponde un inserimento nella coda a priorità e al carattere \* un'estrazione con cancellazione del massimo. Si riporti la configurazione della coda a priorità dopo ogni operazione e la sequenza dei valori restituiti dalle estrazioni con cancellazione del massimo. Al termine si cambi la priorità di 38 in 11 e si disegni la configurazione risultante della coda a priorità.

## 4. (1.5 punti)

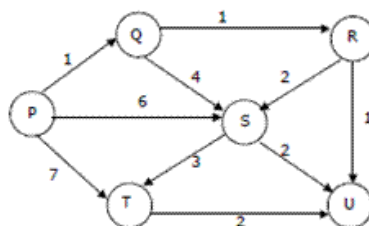
Si esprima in notazione infissa, prefissa e postfissa la seguente espressione aritmetica mediante visita dell'albero binario corrispondente:  $A * ((B / C) - (D * (E - F)))$

## 5. (2 punti)

Sia data la sequenza di chiavi GFERBAOI, dove ciascun carattere è individuato dal suo ordine progressivo nell'alfabeto ( $A=1, \dots, Z=26$ ). Si riporti la struttura di una tabella di hash di dimensione 17, inizialmente supposta vuota, in cui avvenga l'inserimento della sequenza indicata. Si supponga di utilizzare l'open addressing con quadratic probing. Si selezionino opportuni valori per  $c_1$  e  $c_2$ .

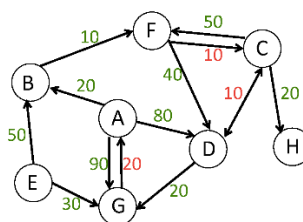
## 6. (2.5 punti)

Sia dato il seguente DAG pesato: considerando **P** come vertice di partenza, si determinino i cammini **massimi** tra **P** e tutti gli altri vertici. Qualora necessario, si trattino i vertici secondo l'ordine alfabetico e si assuma che la lista delle adiacenze sia anch'essa ordinata alfabeticamente.



## 7. (2 punti)

Sia dato il seguente grafo orientato pesato:



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.

# 03MNO Algoritmi e Programmazione

## Appello del 26/06/2017 - Prova di programmazione (18 punti)

Una catena di alberghi gestisce  $N$  hotel in una città dove per un numero noto  $X$  di giorni si svolge un importante evento. Ciascun albergo  $H_i$  ha un numero di camere  $M_i$ . Tramite il suo sito Internet la catena riceve le prenotazioni dei clienti, trascritte su di un file una per riga. Il numero di prenotazioni non è noto a priori. Il formato del file è

`id_cliente arrivo pernottamenti`

dove `id_cliente` è una stringa di al massimo 20 caratteri, `arrivo` è il giorno di arrivo (intero tra 0 e  $X-1$ ) e `pernottamenti` è il numero di notti trascorse in quell'albergo (intero tra 1 e  $X$ ). Si assuma che i clienti siano distinti e che i dati del file siano corretti.

La catena vuole allocare i clienti tra i suoi alberghi:

- tenendo conto delle richieste di ciascun cliente
- rispettando per ogni giorno il vincolo di capacità  $M_i$  di ciascun albergo
- ottimizzando l'utilizzo delle strutture sulla base delle formule seguenti:

il riempimento medio  $r_i$  dell'albergo  $H_i$  nel periodo in questione è definito come l'occupazione delle camere su camere totali disponibili per i giorni ( $0 \leq r_i \leq 1$ ):

$$r_i = \frac{1}{M_i \cdot X} \sum_{j=0}^{X-1} camere_occupate_j$$

siano  $r_{max}$  e  $r_{min}$  rispettivamente il massimo e il minimo riempimento tra i valori di  $r_i$  di tutti gli alberghi. Si massimizzi la funzione:

$$f = a \sum_{i=0}^{N-1} r_i - b \cdot \frac{r_{max} - r_{min}}{r_{max}}$$

dove  $a$  e  $b$  sono valori interi positivi ricevuti in input.

Si scriva un programma C che, ricevuti sulla riga di comando:

- come primo argomento il nome del file contenente le prenotazioni
- come secondo argomento il nome di un file che contiene una possibile allocazione dei clienti uno per riga nella forma  
`id_cliente albergo`  
dove `albergo` è un intero tra 0 e  $N-1$
- come terzo argomento il nome di un file di output

letti da input gli interi  $N$ ,  $X$ ,  $a$  e  $b$  e il vettore  $M$

- verifichi se l'allocazione proposta nel secondo file rispetti i vincoli di capacità giornaliera (non tenendo conto quindi dei criteri di ottimalità)
- determini, se esiste, un'allocazione dei clienti tra gli alberghi
  - tenendo conto delle richieste di ciascun cliente
  - rispettando per ogni giorno il vincolo di capacità  $M_i$  di ciascun albergo
  - massimizzando la funzione  $f$

e la scriva sul file di output nel formato (per ogni riga) `id_cliente albergo`.

### PER ENTRAMBE LE PROVE DI PROGRAMMAZIONE (18 o 12 punti):

- indicare nell'elaborato e nella relazione (oltre a nome, cognome e numero di matricola) anche il nome del corso per cui si sta sostenendo l'esame (AP, APA I+II).
- Se non indicato diversamente, è consentito utilizzare chiamate a funzioni standard, quali ordinamento per vettori, funzioni su FIFO, LIFO, liste, BST, tabelle di hash, grafi e altre strutture dati, considerate come librerie esterne.
- Gli header file devono essere allegati all'elaborato (il loro contenuto riportato nell'elaborato stesso). Le funzioni richiamate, inoltre, dovranno essere incluse nella versione del programma allegata alla relazione. I modelli delle funzioni ricorsive non sono considerati funzioni standard.
- Consegna delle relazioni (per entrambe le tipologie di prova di programmazione): entro giovedì 29/06/2017, alle ore 23:59, mediante caricamento su Portale. Le istruzioni per il caricamento sono pubblicate sul Portale nella sezione Materiale). **QUALORA IL CODICE CARICATO CON LA RELAZIONE NON COMPILI CORRETTAMENTE, VERRÀ APPLICATA UNA PENALIZZAZIONE.** Si ricorda che la valutazione del compito viene fatta senza discussione o esame orale, sulla base dell'elaborato svolto in aula. Non verranno corretti i compiti di cui non sarà stata inviata la relazione nei tempi stabiliti.

# 03MNO Algoritmi e Programmazione

## Appello del 26/06/2017 - Prova di programmazione (12 punti)

### NON E' AMMESSO L'USO DI FUNZIONI DI LIBRERIA

#### 1. (2 punti)

Si scriva una funzione con il seguente prototipo

```
float my_atof(char *s);
```

che riceve una stringa di caratteri che rappresenta un numero reale e ritorna il corrispondente valore float. Nella stringa sono ammessi solo i caratteri '+', '-', '.', '0' e '9' secondo il seguente formato:

- il primo carattere è il segno positivo o negativo
- segue la parte intera del numero come numero imprecisato di caratteri tra '0' e '9' terminato da '.'
- segue infine la parte decimale del numero come numero imprecisato di caratteri tra '0' e '9'.

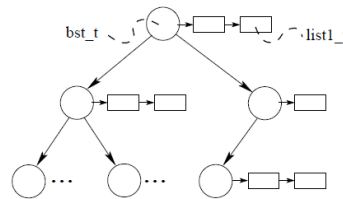
Si assuma che la stringa in ingresso soddisfi sempre il formato sopra descritto. Esempi di stringhe corrette sono: +123.45, -3.5678, -678.2.

#### 2. (4 punti)

Un database di modelli di automobile è organizzato come BST. Ogni nodo del BST contiene le seguenti informazioni:

- nome del modello (stringa di caratteri che funge da chiave per il BST)
- puntatore a una lista non ordinata i cui nodi contengono le seguenti informazioni:
  - nome del concessionario (stringa di caratteri)
  - prezzo in Euro (float).

I nodi del BST sono di tipo `bst_t` quelli della lista di tipo `list1_t`. La figura seguente rappresenta graficamente un esempio di database:



Si scriva una funzione con il seguente prototipo

```
void bst2list(bst_t *bst, list2_t **head, list2_t **tail);
```

che riceve come parametri un puntatore alla radice del BST e i puntatori alla testa e alla coda di una seconda lista. Si crei una lista ordinata per modelli (ordine alfabetico ascendente), i cui nodi contengono il nome del modello, il nome del concessionario che lo vende a prezzo minimo e il prezzo minimo. L'inserzione in detta lista avvenga in coda. Si riportino esplicitamente le definizioni dei tipi `bst`, `list1` e `list2`.

#### 3. (6 punti)

Siano una matrice di interi `mat`, le sue dimensioni `r` e `c` e un intero `k` i parametri della funzione `f` con il seguente prototipo:

```
void f(int **mat, int r, int c, int k);
```

La funzione deve trovare e stampare (se esiste) uno qualsiasi dei sottoinsiemi dei valori della matrice:

- di cardinalità minima
- di cardinalità massima

la somma dei cui elementi sia `k`.

Esempio:

`r = 3, c = 4, k = 10` e `mat`

	0	1	2	3
0	2	3	1	2
1	6	4	2	5
2	2	4	3	2

uno dei sottoinsiemi a cardinalità minima la cui somma vale 10 è (6, 4), uno dei sottoinsiemi a cardinalità massima la cui somma vale 10 è (2, 3, 1, 2, 2).