

Esame del 22 maggio 2003

Parte I (teoria)

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 21 37 45 20 30 18 22 40 36 65

Punto 1.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 (sotto forma di vettore). Si ipotizzi che, alla fine, nella radice dell'heap sia memorizzato il valore minimo.

Punto 1.b (2 punti)

Si eseguano su tale heap i primi 3 passi dell'algoritmo di heapsort, riportando il contenuto del vettore dopo ciascun passo dell'algoritmo.

Esercizio 2 (3 punti)

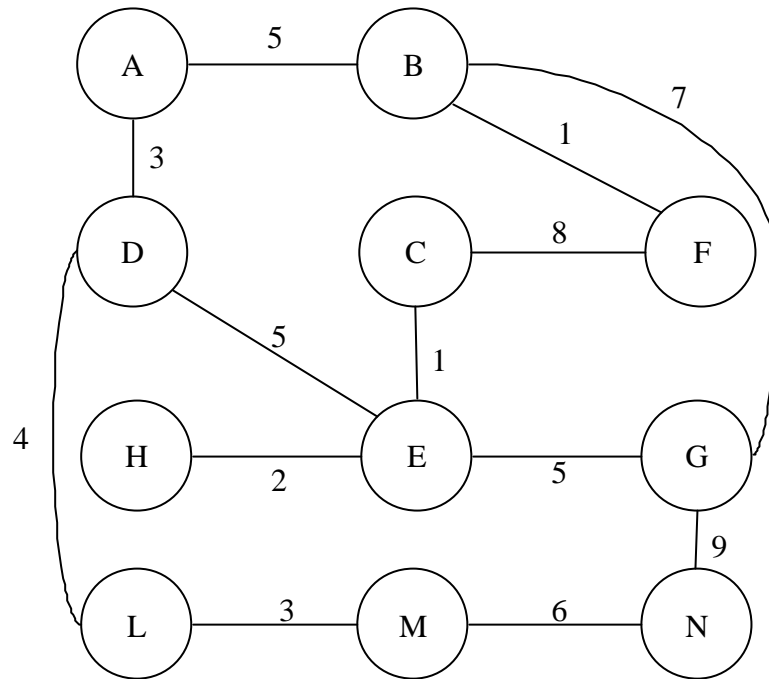
Si consideri una tabella di hash basata su open addressing che utilizza la tecnica del linear probing per risolvere le collisioni. La tabella è inizialmente vuota. Su di essa viene eseguita la seguente sequenza di operazioni di inserimento di elementi con chiave corrispondente ad un numero:

48 56 75 32 18 94 45 23 20 79 78 77 41 42 43 33 12 46 38

Si mostri il contenuto della tabella di hash al termine della sequenza di inserimenti, assumendo che la sua dimensione sia pari a $M=21$ e che la funzione di hash utilizzata corrisponda al resto della divisione della chiave per M .

Esercizio 3

Sia dato il seguente grafo:



Punto 3.a (4 punti)

Si visiti il grafo in *profondità* ed in *ampiezza*, considerando A come vertice di partenza. Si indichi la sequenza di visita dei vertici nei due casi. Qualora necessario, si trattino i vertici secondo l'ordine alfabetico.

Punto 3.b (3 punti)

Si calcoli un albero di copertura minimo del grafo utilizzando l'algoritmo di Kruskal.

Parte II (programmazione)

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

Nel Medio Evo le comunicazioni avvenivano tramite segnalazioni luminose da un castello ad un altro. Si supponga di conoscere la mappa dei castelli e, per ogni castello, l'elenco dei castelli vicini raggiungibili a partire da questo tramite segnalazioni ottiche.

Si scriva un programma che permetta di determinare se, dato un castello di partenza, sia possibile comunicare con un castello destinazione, eseguendo un numero arbitrario di passaggi.

Si scriva un programma che esegua le seguenti operazioni:

- acquisisce da un file (il cui nome va letto da tastiera) le informazioni relative ai castelli. Il file contiene nella prima riga il numero n dei castelli ed il numero m delle righe successive; in ciascuna delle m righe successive vi è una coppia del tipo

```
<numero_castello1> <numero_castello2>
```

che indica che il `castello1` può comunicare con il `castello2`. Sulla base delle informazioni acquisite, il programma costruisce in memoria un'adeguata rappresentazione.

- leggere da tastiera il numero del castello di origine della comunicazione e quello del castello destinazione.
- verifica se è possibile inviare un messaggio dal castello di origine a quello destinazione, passando attraverso un numero arbitrario di castelli intermedi, visualizzando un messaggio adeguato nei due casi.