

Cognome e Nome _____

Matricola _____

Appello del 2 luglio 2018

Esercizio 1 (8 punti)

Si consideri il seguente algoritmo di ricerca binaria ricorsiva:

```
function binarySearchRic(int[] A, int p, int r, int v)
    if p > r
        return -1
    if v < A[p] or v > A[r]
        return -1
    q=p+((r-p)/2)
    if A[q] == v
        return q
    if A[q] > v
        return binarySearchRic(A, p, q-1, v)
    else
        return binarySearchRic(A, q+1, r, v)
```

Scrivere la ricorrenza $T(n)$ che indica il tempo di esecuzione dell'algoritmo nel caso peggiore. Si dia una stima esplicita (non ricorsiva) di $T(n)$ facendo uso di un metodo a scelta.

Esercizio 2 (8 punti)

Si scriva un algoritmo (in pseudocodice o in Java) che, presi in input due array **a1** e **a2** di numeri interi, restituisca il numero degli elementi di **a1** che non sono presenti in **a2** (si può assumere che né **a1** né **a2** contengano elementi ripetuti) e **se ne analizzi la complessità computazionale**. Tale algoritmo non deve modificare gli array ricevuti in input, può richiamare qualsiasi algoritmo visto a lezione e deve avere complessità temporale **$O((n_1 + n_2) \log n_2)$** nel caso peggiore (*algoritmi con complessità temporale peggiore danno luogo a una valutazione minore, pari a un massimo 4 punti*), dove n_1 e n_2 sono le lunghezze dei due array in input **a1** e **a2**, rispettivamente.

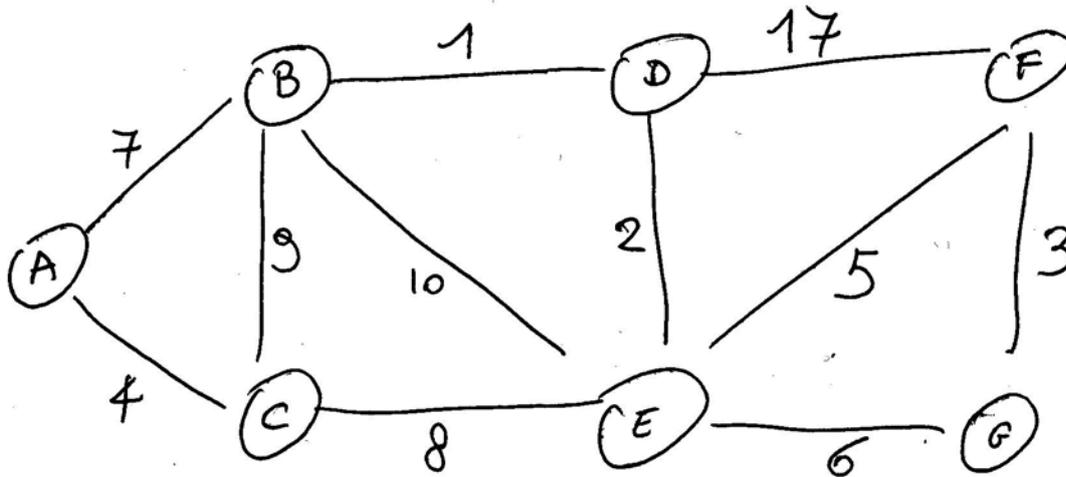
Se ad esempio gli array sono {1, 3, 9, 5, 7} e {10, 1, 7, 8, 4}, l'algoritmo deve restituire **3** in quanto gli elementi 3, 9 e 5 del primo array non sono presenti nel secondo array.

Esercizio 3 (8 punti)

- a. Mostrare l'**heap di minimo** che si ottiene a partire dall'heap vuoto inserendo nell'ordine indicato le chiavi **50, 32, 84, 5, 2, 8, 90, 76, 43** (mostrare l'heap che si ottiene dopo l'inserimento di ogni chiave)
- b. Rimuovere per 3 volte la chiave minima dall'heap, mostrando l'heap che si ottiene dopo ogni estrazione
- c. Inserire nell'heap così ottenuto le chiavi **35, 27, 6, 88**

Esercizio 4 (12 punti)

Si consideri il grafo non diretto in figura.



- Mostrare la rappresentazione del grafo tramite liste di adiacenza
- Mostrare la rappresentazione del grafo tramite matrice di adiacenza
- Si consideri il semplice problema di rispondere alla domanda: *Esiste e che peso ha l'arco tra due nodi dati in input?* Discutere quale rappresentazione del grafo rende possibile risolvere tale problema in tempo costante nel caso peggiore (non dipendente dalla dimensione del grafo).
- Eseguire **passo passo** la visita in ampiezza a partire dalla sorgente **F**, indicando l'ordine di visita dei nodi e le distanze calcolate dalla sorgente.
- Eseguire **passo passo** la visita in profondità a partire dalla sorgente **B**, indicando il tempo di inizio e fine visita per ogni nodo.

Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **100** minuti.
- Scrivere **subito** nome, cognome, matricola e numero del compito su **OGNI FOGLIO (compreso questo)**.
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- È possibile consultare appunti, libri, dispense o qualsiasi altro materiale.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc...) deve essere **completamente disattivato** e **depositato in vista sulla cattedra**
- Mettere in vista sul banco il proprio libretto (o altro documento di identità).