

Esame di algoritmi e strutture dati

17 gennaio 2019

Tempo a disposizione: 2 ore

Esercizio 1

(7 punti)

1. Mostrare lo pseudocodice dell'algoritmo `IntegerSort`.
2. Determinare, fornendo una dimostrazione formale, il costo temporale dell'algoritmo nel caso peggiore.
3. Mostrare i passi eseguiti dall'algoritmo sul seguente array di input:

8	7	8	5	1	2	4	4
---	---	---	---	---	---	---	---

Esercizio 2

(7 punti)

Si considerino i seguenti due algoritmi che, data una lista l di interi, restituiscono la somma dei valori in essa contenuti:

```
Intero SommaIterativa(Lista l){  
   $n = |l|;$   
   $r = 0;$   
  foreach( $i = 0, \dots, n - 1$ )  $r += l[i];$   
  return r;  
}
```

```
Intero SommaRicorsiva(Lista l){  
   $n = |l|;$   
  if( $n == 0$ ) return 0;  
  return  $l[0] + \text{SommaRicorsiva}(l[1, n - 1]);$   
}
```

dove:

- $|l|$ rappresenta la dimensione della lista l ;
- il primo elemento di una lista ha indice 0;
- $l[i, j] = \langle l[i], l[i + 1], \dots, l[j] \rangle$.

1. Determinare, motivando la risposta, il costo temporale di ciascun algoritmo;
2. Determinare, motivando la risposta, il costo spaziale di ciascun algoritmo;
3. Discutere, in al più due frasi, se e quale degli algoritmi sia preferibile all'altro.

Esercizio 3

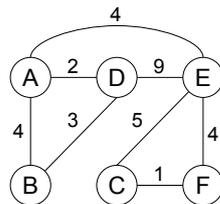
(6 punti)

1. Fornire la definizione di *albero AVL*.
2. Indicare il costo della ricerca di un elemento nel caso peggiore.
3. Mostrare i passi di costruzione di un albero AVL quando vengono effettuate, nell'ordine riportato, le seguenti operazioni:
 - inserimento di un nodo con chiave 20;
 - inserimento di un nodo con chiave 8;
 - inserimento di un nodo con chiave 9;
 - inserimento di un nodo con chiave 6;
 - inserimento di un nodo con chiave 2;
 - eliminazione del nodo con chiave 6;

Esercizio 4

(5 punti)

1. Fornire la definizione di *minimo albero ricoprente*;
2. Mostrare i passi eseguiti dall'algoritmo di Kruskal quando eseguito sul seguente grafo di input:



Esercizio 5

(5 punti)

Progettare un algoritmo (pseudocodice) con segnatura:

$$\text{Intero Altezza}(\text{AlberoBinario } T)$$

che, preso in input un albero binario T , ne restituisce l'altezza.

Esercizio 6

(3 punti)

ATTENZIONE: esercizio valido solo se con gli esercizi 1–5 viene raggiunto il punteggio minimo di 20.

Descrivere a quale scopo servono gli algoritmi di Bellman-Ford e Dijkstra ed illustrare se uno dei due sia preferibile rispetto all'altro, ed in quali condizioni.