

## Esercizi per la preparazione allo scritto di Algoritmi e Strutture Dati (Modulo A)

1. Dimostrare o argomentare la scorrettezza delle seguenti affermazioni:

A. Poiché  $n=O(n)$ ,  $2n=O(n)$ ,  $3n=O(n)$ , ecc., allora  $\sum_{k=1}^n kn = \sum_{k=1}^n O(n)$

B.  $\sum_{k=1}^n O(n) = O(n^2)$

2. Fornire una dimostrazione del fatto che per ogni  $m$ :

$$\lim_{n \rightarrow \infty} \frac{(\log n)^m}{n} = 0$$

3. Valutare la seguente sommatoria:

$$\sum_{k=1}^n (2k - 1)$$

4. Valutare la seguente sommatoria:

$$\sum_{k=1}^{\log \log n} 2^k$$

5. Utilizzando la definizione della notazione asintotica, dimostrare o fornire una contro prova delle seguenti:

A.  $a n \log n + 3 n = \Theta(n \log n)$

B.  $2a \sqrt{n} \log n + 7b n = O(n \log n)$

C.  $a n \log n + 3 n = \Theta(\sqrt{n} \log n)$

6. Calcolare la soluzione asintoticamente migliore della seguente ricorrenza:

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ 2T(\lfloor n/2 \rfloor + 5) + n & \text{se } n \geq 2 \end{cases}$$

7. Risolvere con il Metodo Iterativo prima e, ove possibile, con il Metodo Principale poi, ciascuna delle ricorrenze del problema 4-1 (capitolo 4 del Cormen). Dimostrare per induzione la correttezza dei risultati ottenuti, (cioè se risulta che la soluzione della ricorrenza è la funzione  $f(n)$ , dimostrare che  $T(n)=\Theta(n)$  ).
8. Scrivere una versione iterativa dell'algoritmo Heapify.
9. Come dovrebbero essere modificati gli algoritmi discussi per HeapSort in modo da ordinare una sequenza in ordine decrescente (dal più grande al più piccolo)?
10. Problema 8-1 (capitolo 8 del Cormen)
11. Problema 8-3 domanda *b*) (capitolo 8 del Cormen)
12. Esibire una sequenza di 10 interi che determinano il caso peggiore per l'algoritmo QuickSort, esibire una sequenza della stessa lunghezza che ne determina il caso migliore.
13. Esibire una sequenza di 10 interi che determinano il caso peggiore per l'algoritmo QuickSort Randomizzato; esibire una sequenza della stessa lunghezza che ne determina il caso migliore.
14. Esercizio 8.1-4 (capitolo 8 del Cormen)
15. Spiegare dettagliatamente come modificare l'algoritmo di QuickSort in modo che il tempo di esecuzione sia  $O(n \log n)$  anche nel caso peggiore.
16. È possibile stampare in ordine crescente gli elementi di un albero binario di ricerca in tempo  $O(n)$ ? Se sì descrivere un algoritmo, altrimenti spiegare perché non è possibile.
17. È possibile stampare in ordine crescente gli elementi di uno Heap in tempo  $O(n)$ ? Se sì descrivere un algoritmo, altrimenti spiegare perché non è possibile.
18. Data una sequenza arbitraria di  $n$  numeri interi. Qual'è il limite inferiore del tempo di esecuzione di un algoritmo che costruisce l'albero di ricerca binario con gli elementi della sequenza (suggerimento: pensate a come può essere usata la proprietà di un albero binario di ricerca per realizzare un algoritmo di ordinamento).
19. Qual'è il numero minimo di nodi non *Nil* di un albero Red-Black di altezza nera  $k$ ? Qual'è il numero massimo?
20. Dimostrare che gli algoritmi di rotazione in alberi binari di ricerca preservano sempre l'ordinamento in ordine dei nodi di un albero binario.