

“ALGORITMI”
CORSO DI STUDIO IN INFORMATICA (laurea triennale)
UNIVERSITÀ DEGLI STUDI DI CATANIA
ANNO ACCADEMICO 2014/15

Prima sessione di esami (I appello) - 09 febbraio 2015

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

ESERCIZIO 1

Si risolva l'equazione di ricorrenza

$$T(n) = 16 \cdot T\left(\frac{n}{a^2}\right) + \Theta(n)$$

al variare del parametro reale $a > 1$.

ESERCIZIO 2

- (a) Si illustri la struttura dati del max-heap binario mettendola anche in relazione con la sua rappresentazione con array.
- (b) Si descrivano le procedure MAX-HEAPIFY e BUILD-MAX-HEAP e si illustri l'azione di BUILD-MAX-HEAP sulla sequenza di interi [1, 3, 9, 1, 1, 10, 12, 2, 14, 2].
- (c) Si descriva l'algoritmo Heapsort.

ESERCIZIO 3

- (a) Si illustri la struttura dati degli alberi “rosso-neri”.
- (b) Si enunci una minorazione del numero di nodi interni in un sottoalbero radicato in un nodo x di un albero rosso-nero e la si utilizzi per dimostrare un limite superiore all'altezza di un albero rosso-nero con n nodi interni.
- (c) Si illustri l'inserimento delle chiavi 38, 35, 21, 10, 12, 4 in un albero rosso-nero inizialmente vuoto.

ESERCIZIO 4

Sia \otimes un'operazione *associativa* su matrici di numeri reali tale che, date due matrici A e B rispettivamente di dimensioni $p \times q$ e $q \times r$, produce una matrice $A \otimes B$ di dimensione $p \times r$, effettuando $p^2q^2 + r^3$ operazioni elementari.

Sia $\mathcal{A} = (A_1, A_2, \dots, A_n)$ una sequenza di matrici di dimensioni $p_{i-1} \times p_i$, per $i = 1, 2, \dots, n$.

Utilizzando la metodologia della programmazione dinamica, si descriva un'algoritmo per determinare la parentizzazione della sequenza \mathcal{A} che consenta di calcolare la matrice

$$A_1 \otimes A_2 \otimes \dots \otimes A_n$$

con il minor numero possibile di operazioni elementari.

Qual è la complessità dell'algoritmo trovato in funzione della lunghezza n della sequenza \mathcal{A} ?

ESERCIZIO 5

Nel contesto della metodologia *greedy*, si enunci il problema di ottimizzazione relativo alla *selezione di attività* e se ne discuta una soluzione efficiente, valutandone la complessità computazionale e illustrandola sul seguente insieme $S = \{a_1, \dots, a_{10}\}$ di attività, caratterizzate dai seguenti tempi iniziali e finali:

i	1	2	3	4	5	6	7	8	9	10
s_i	11	13	7	2	1	4	12	5	10	6
f_i	12	14	9	5	6	7	13	10	12	9