

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

Terza sessione di esami - 21 settembre 2015

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

ESERCIZIO 1 (Equazione di ricorrenza)

Si enuncino il Teorema Master ed il suo Corollario, quindi si risolva la seguente equazione di ricorrenza al variare del parametro $\alpha > 1$:

$$T(n) = \alpha \cdot T\left(\frac{n}{3}\right) + n^2 \log n.$$

Per quali valori di α si ha: (a) $T(n) = \Theta(n^3)$; (b) $T(n) = \Theta(n^2 \log^2 n)$?

ESERCIZIO 2 (Ordinamento)

Si descrivano la struttura dati MAX-HEAP e le procedure MAX-HEAPIFY, BUILD-MAX-HEAP e HEAPSORT, determinandone le complessità computazionali.

ESERCIZIO 3

Sia dato il grafo non orientato \mathcal{G} rappresentato dalle seguenti liste di adiacenza

A \rightarrow B, C, E	D \rightarrow A	G \rightarrow A, E
B \rightarrow A	E \rightarrow B	H \rightarrow G, I
C \rightarrow D, E	F \rightarrow G, H	I \rightarrow F

Dopo aver descritto l'algoritmo di visita in profondità, si effettui la visita in profondità del grafo \mathcal{G} a partire dal vertice A, indicando per ogni vertice i tempi di inizio e fine visita, e la classificazione di tutti gli archi (es. archi d'albero, all'indietro, ecc.).

ESERCIZIO 4 (Tavole hash)

(a) Data la funzione $h(x, i) =_{Def} (x + 3i) \bmod 17$, si illustri l'inserimento delle chiavi

23, 43, 21, 5, 62, 72, 58, 48, 52, 46, 78, 55, 35, 17, 51

in una tabella hash di dimensione 17, inizialmente vuota e organizzata con l'indirizzamento aperto, utilizzando $h(x, i)$ come funzione hash.

(b) Si enunci l'ipotesi di *hashing uniforme*, si forniscano dei limiti superiori al numero medio di scansioni in ricerche *con* e *senza* successo in una tabella hash con fattore di carico α , assumendo l'ipotesi di hashing uniforme.

ESERCIZIO 5 (Algoritmi greedy)

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	4	1	2	12	5	7	13	10	6
f_i	12	7	6	5	13	10	9	14	12	9