

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

Seconda sessione di esami (I appello) - 22 giugno 2015

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

**ESERCIZIO 1 (Equazione di ricorrenza)**

Si risolva la seguente equazione di ricorrenza parametrica, al variare del parametro reale  $\beta > 1$ ,

$$T(n) = 9 \cdot T\left(\frac{n}{\beta}\right) + n^2 \log n,$$

e quindi si determini per quali valori di  $\beta$  si ha: (a)  $T(n) = \Theta(n^2)$ ; (a)  $T(n) = \mathcal{O}(n^2)$ .

**ESERCIZIO 2 (Ordinamento)**

Si descriva l'algoritmo COUNTING-SORT (con pseudocodice), nonché il suo ambito di applicabilità. Quindi si determini la sua complessità computazionale.

**ESERCIZIO 3 (Programmazione dinamica)**

Si consideri la seguente operazione  $\oplus$  sui numeri naturali, definita da:  $a \oplus b =_{Def} 2a + 3b$ .

- (a) Si verifichi con un esempio a scelta che l'operazione  $\oplus$  non è associativa.
- (b) Utilizzando la metodologia della programmazione dinamica, si determini un algoritmo che, data una sequenza di numeri naturali  $a_1, a_2, \dots, a_n$ , calcoli il valore *massimo* che l'espressione  $a_1 \oplus a_2 \oplus \dots \oplus a_n$  possa assumere al variare di tutte le possibili parentesizzazioni.  
Si determini la complessità computazionale dell'algoritmo ottenuto.

**ESERCIZIO 4 (Visita in profondità)**

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

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

Dopo aver descritto l'algoritmo di visita in profondità, lo si utilizzi per visitare il grafo  $\mathcal{G}$  a partire dal vertice A, rappresentando la foresta DFS ottenuta e indicando per ogni vertice i tempi di inizio e di fine visita.

Utilizzare i risultati della visita effettuata per stabilire se il grafo  $\mathcal{G}$  è aciclico.

**ESERCIZIO 5 (Tavole hash)**

- (a) Data la funzione  $h(x, i) =_{Def} (x + 3i) \bmod 19$ , si illustri l'inserimento delle chiavi

69, 12, 34, 8, 33, 2, 32, 65, 59, 42, 70, 60, 51, 16, 15

in una tabella hash di dimensione 19, 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  $\alpha$ , assumendo l'ipotesi di hashing uniforme.