

Nome: _____ Cognome: _____

Matricola: _____ Firma: _____

Università di Salerno

31 gennaio 2013

Algoritmi
(Matricole congrue ad 1 mod 3)

Docente: Alfredo De Santis

Durata: 2 ore

Nessun materiale ammesso per consultazione. Buon lavoro a tutti.

Il presente esame consiste di **6** pagine e **5** quesiti. Segnalare qualsiasi discrepanza alla commissione. Il numero in parentesi all'inizio di ciascun quesito corrisponde al numero di punti assegnati ad una risposta corretta.

Rispondere a tutti i quesiti.

Riservato alla commissione:

	Punti
1 (13)	
2 (20)	
Totale Parziale	

	Punti
3 (21)	
4 (21)	
5 (25)	
Totale Parziale	

Totale (100)	
--------------	--

1. [13] NOTAZIONI ASINTOTICHE.

Per ognuna delle seguenti coppie di funzioni indicare che relazione vale tra $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, $f(n) = \Theta(g(n))$ (E' necessario motivare le risposte):

(a) $f(n) = \log n^2$; $g(n) = \log n + 5$

(b) $f(n) = n$; $g(n) = \log n^2$

(c) $f(n) = \log \log n$; $g(n) = \log n$

(d) $f(n) = 10$; $g(n) = \log 10$

Lo spazio per la risposta continua sulla prossima pagina

2. [20] BREADTH-FIRST SEARCH.

Si descriva ed analizzi la Breadth-First Search. In particolare si chiariscano i seguenti aspetti: layer, relazioni tra archi del grafo e layer, implementazione ed analisi, relazioni con componenti connesse, e relazioni con grafi bipartiti.

Lo spazio per la risposta continua sulla prossima pagina

3. [21] CONTEGGIO INVERSIONI.

Si enunci il problema chiarendo quali sono gli input e quali gli output e che cosa é una inversione. In seguito, si descriva ed analizzi un algoritmo che risolve il problema del conteggio delle inversioni.

Lo spazio per la risposta continua sulla prossima pagina

4. [21] INTERVAL PARTITIONING.

Si definisca il problema chiarendo quali sono gli input e gli output. Si descriva ed analizzi l'algoritmo, provando la correttezza della scelta greedy e poi discutendo l'implementazione ed il running time.

Lo spazio per la risposta continua sulla prossima pagina

5. [25] Si descriva ed analizzi un algoritmo per la seguente variazione del problema dello zaino: Dati n oggetti di peso w_1, w_2, \dots, w_n e valore v_1, v_2, \dots, v_n ed uno zaino di capacità W (tutti gli input sono > 0), trovare il massimo valore di un sottoinsieme degli oggetti il cui peso totale é $\leq W$, con la condizione che ogni oggetto pu essere preso al massimo 2 volte. (La variazione rispetto al problema del testo, consiste nel superamento del vincolo che ogni oggetto poteva essere preso al massimo una sola volta.)