

Cognome:

Nome:

Matricola:

Rispondere alle domande usando lo spazio designato. Gli appunti vanno scritti ESCLUSIVAMENTE nelle pagine finali. Non usare altri fogli.

Indicare eventuale preferenza per la data dell'orale:

Prima del 12 luglio	Luglio (dopo il 12)	Settembre

1	2	3	4	5	6	Tot/100
/16	/17	/16	/17	/17	/17	/100

1. 16 punti

Si considerino le seguenti funzioni: $4\sqrt{n} + \log n$, $\log \log n$, 2^n , $n^{\log n}$, $13n^3$, $n + 15$.

a) Si ordinino le funzioni scrivendole da sinistra a destra, in modo tale che la funzione $f(n)$ sia posta a sinistra di $g(n)$ se $f(n) = O(g(n))$.

b) Si dimostri formalmente (cioè fornendo le costanti) almeno due (a scelta) dei confronti affermati al punto a). In altre parole se l'ordine proposto è: $f_1(n)$, $f_2(n)$, $f_3(n)$, $f_4(n)$, $f_5(n)$, $f_6(n)$, allora occorre dimostrare che $f_i(n) = O(f_{i+1}(n))$ per almeno due diversi indici i .

2. *17 punti*

L'algoritmo 3-MERGESORT e' una modifica dell'algoritmo MERGESORT che opera come segue. Dato un array $A[1..n]$ divide A in 3 sottovettori di uguale grandezza; ordina ricorsivamente ogni sottovettore; chiama la procedura MERGE sui primi due vettori ed infine chiama la procedura MERGE sul vettore cosi' ottenuto e sul terzo sottovettore.

- a) Scrivere lo pseudocodice dell'algoritmo 3-MERGESORT.
- b) Si analizzi la complessita' dell'algoritmo proposto, nell'ipotesi che n sia una potenza di 3.

3. *16 punti*

- a) Si disegni l'albero binario di ricerca ottenuto dopo l'inserimento in ordine in un albero inizialmente vuoto delle seguenti chiavi: 9, 7, 12, 3, 8, 10.
- b) Si colori l'albero in modo da ottenere un albero rosso-nero e si indichi l'altezza nera di ogni nodo.
- c) Si applichi la RIGHT-ROTATE intorno al nodo con chiave 7.
- d) Si colori l'albero in modo da ottenere un albero rosso-nero o si spieghi perche' cio' non e' possibile.

4. 17 punti

Si consideri il problema di determinare il minimo numero di monete da 1, 5 e 10 centesimi necessarie per scambiare un ammontare di denaro D , supponendo di poter utilizzare un numero illimitato di monete dello stesso taglio.

- a) Descrivere ed analizzare un algoritmo goloso (*greedy*) che lo risolve.
- b) Enunciare e dimostrare la proprieta' della scelta golosa (*greedy*) per l'algoritmo proposto.
- c) Spiegare perche' la vostra dimostrazione del punto b) non varrebbe per il problema dello scambio con monete da 1, 7 e 10 centesimi.

5. 17 punti

Dati gli interi $0 \leq j \leq i$, il numero di Stirling del secondo tipo, denotato $S(i, j)$, e' definito da:

$$S(i, j) = jS(i-1, j) + S(i-1, j-1)$$

$$S(i, 0) = 0 \text{ se } i \geq 1 \text{ e } S(i, i) = 1 \text{ per ogni } i \geq 0.$$

Descrivere ed analizzare un algoritmo di programmazione dinamica che, dato un intero $n \geq 0$ restituisca il numero $S(n, n-1)$. E' necessario scrivere lo pseudocodice ed analizzarne il tempo di esecuzione.

6. 17 punti

Una k -colorazione di un grafo non orientato $G = (V, E)$ e' una funzione $c : V \longrightarrow \{1, 2, \dots, k\}$, che assegna ad ogni vertice $v \in V$ un colore $c(v)$ in modo che vertici adiacenti abbiano colori distinti (formalmente se $(u, v) \in E$ allora $c(u) \neq c(v)$). Il *problema della colorazione* di un grafo e' quello di determinare il numero minimo di colori richiesti per una colorazione del grafo.

- a) Si scriva il linguaggio COLOR associato a tale problema.
- b) Si dimostri che $\text{COLOR} \in NP$.

PAGINA PER APPUNTI O 'BRUTTA COPIA'

PAGINA PER APPUNTI O 'BRUTTA COPIA'