

Algoritmi e Strutture Dati

a.a. 2009/10

Compito del 5/2/2010

Cognome: _____ Nome: _____

Matricola: _____ E-mail: _____

Parte I

(30 minuti; ogni esercizio vale 2 punti)

1. Completare la seguente tabella indicando la complessità delle operazioni che si riferiscono a un dizionario di n elementi. Si noti che tutte le operazioni, tranne la **Ricerca**, assumono di aver già raggiunto l'elemento x a cui si applica l'operazione.

	Ricerca	Cancellazione	Successore	Costruzione
Lista doppia non ordinata				
Array ordinato				
Alberi binari di ricerca				

2. Risolvere le seguenti relazioni di ricorrenza (giustificando la risposta):

(a) $T(n) = 4T(n/2) + n$

(b) $T(n) = 4T(n/2) + n^2$

(c) $T(n) = 4T(n/2) + n^3$

3. Si definisca la relazione di "riducibilità polinomiale" tra problemi (\leq_p) e si stabilisca se valgono le seguenti proprietà: a) riflessiva, b) simmetrica, c) transitiva (giustificando tecnicamente le risposte).

Algoritmi e Strutture Dati

a.a. 2009/10

Compito del 5/2/2010

Cognome: _____ Nome: _____

Matricola: _____ E-mail: _____

Parte II

(2.5 ore; ogni esercizio vale 6 punti)

1. Progettare un algoritmo che, ricevuto in input un intero k e un array a , **non ordinato**, di n elementi **distinti**, restituisca il k -esimo elemento più piccolo di a .
 - a. Progettare una soluzione di costo in tempo $\Theta(n \log n)$.
 - b. Progettare una soluzione di costo in tempo $O(n + k \log n)$.
2. Un nodo di un albero binario è detto **centrale** se il numero di foglie del sottoalbero di cui è radice è pari alla somma delle chiavi dei nodi appartenenti al percorso dalla radice al nodo stesso.
 - a. Scrivere una funzione **efficiente** in C che restituisca il numero di nodi centrali.
 - b. Discutere la complessità della soluzione trovata.
 - c. Se vogliamo modificare la funzione in modo che restituisca l'**insieme** dei nodi centrali che tipo di struttura dati si può utilizzare per rappresentare l'insieme?
La complessità dell'algoritmo deve rimanere la stessa che nel caso (a).

Si deve utilizzare il seguente tipo per la rappresentazione di un albero binario:

```
typedef struct node{
    int key;
    struct node * left;
    struct node * right;
} * Node;
```

3. Si enunci e si dimostri la proprietà fondamentale degli alberi di copertura minimi e la si utilizzi per dimostrare la correttezza degli algoritmi di Kruskal e Prim.
4. Dato un grafo orientato e pesato $G=(V,E)$ con pesi strettamente positivi, cioè $w(u,v)>0$ per ogni $(u,v) \in E$, si vuole determinare se esiste in G un ciclo $c \equiv \langle x_0, x_1, \dots, x_q \rangle$ raggiungibile da un dato vertice "sorgente" s , in cui il prodotto dei pesi sugli archi sia minore di 1, cioè:

$$\prod_{i=1}^q w(x_{i-1}, x_i) < 1.$$

Si sviluppi un algoritmo per risolvere questo problema, se ne discuta la correttezza e si determini la sua complessità computazionale. Inoltre, si simuli la sua esecuzione sul seguente grafo:

