

Cognome:

Nome:

Matricola:

Riga:

Col:

Algoritmi e Strutture Dati - Prova d'esame
11/01/12

Esercizio 0 Scrivere correttamente nome, cognome, numero di matricola, riga e colonna.

Esercizio 1 - Punti ≥ 4 (Parte A)

Si consideri la seguente ricorrenza

$$T(n) = \begin{cases} T(\lfloor n/c \rfloor) + \Theta(1) & n > 1 \\ 1 & n \leq 1 \end{cases}$$

Si calcoli la corrispondente funzione di complessità utilizzando il metodo di sostituzione.

Esercizio 2 - Punti ≥ 6 (Parte B)

Avete a disposizione n dadi e n bulloni. Tutti i dadi hanno dimensioni diverse; tutti i bulloni hanno dimensioni diverse; per ogni dado, esiste un bullone corrispondente e viceversa. Non potete confrontare due bulloni o due dadi; potete però confrontare un bullone b con un dado d (operazione $\text{try}(b, d)$), e ottenere -1 se il bullone è più piccolo del dado, 0 se coincidono e +1 se il bullone è più grande del dado. Scrivere un algoritmo che trova la coppia bullone e dado più piccoli, discuterne correttezza e complessità misurata nel numero di operazioni $\text{try}()$.

Esercizio 3 - Punti ≥ 10 (Parte A)

Siano X e Y due vettori, ciascuno contenente n interi già ordinati. Descrivete un algoritmo che in tempo $O(\log n)$ trovi la mediana dei $2n$ elementi dei vettori X e Y (cioè dell'insieme $X \cup Y$).

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.

Esercizio 4 - Punti ≥ 14 (Parte B)

Sia k un vettore contenente n chiavi ordinate e distinte e sia p un vettore contenente n pesi, tali per cui $p[i]$ è la frequenza con cui si stima la chiave $k[i]$ verrà ricercata. Supponete di inserire le n chiavi in un albero di ricerca. Il *costo* dell'albero di ricerca per le n chiavi viene definito come

$$\sum_{i=1}^n p[i] \cdot (l[i] + 1)$$

, dove $l[i]$ è il livello dell'albero in cui si trova la chiave $k[i]$.

Scrivere un algoritmo che costruisca e restituisca l'albero binario di ricerca di costo minimo.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.

Suggerimento: si utilizzi la programmazione dinamica.