

Problem Set 2

docente: Luciano Gualà

Esercizio 1 (equazioni di ricorrenza)

Si risolvano le seguenti equazioni di ricorrenza. Si assuma sempre $T(\text{costante}) = O(1)$.

(a) $T(n) = T(n - 10) + 10$.

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

(c) $T(n) = T(n/3) + T(n/6) + n^{\sqrt{\log n}}$.

(d) $T(n) = T(\sqrt{n}) + \Theta(\log \log n)$.

(e) $T(n) = T(n/2 + \sqrt{n}) + \Theta(1)$.

(f) $T(n) = \sqrt{n}T(\sqrt{n}) + n$.

Esercizio 2 (visite di alberi)

Sia T un albero binario di n nodi con radice r . La *profondità* di un nodo v è il numero di archi del cammino da v alla radice. Un nodo u è un *discendente* di v se u si trova nel sottoalbero di T radicato in v . Si assuma che T è mantenuto attraverso una struttura collegata e che ogni nodo v mantenga i puntatori al padre e ai figli ($v.p$, $v.s$, $v.d$). Si progetti un algoritmo con complessità temporale $O(n)$ che, preso T , restituisca il nodo v di profondità minima la cui profondità è maggiore o uguale al numero dei suoi discendenti. Si fornisca lo pseudocodice dettagliato dell'algoritmo e possibilmente non si usino variabili globali e passaggi di parametri per riferimento.

Esercizio 3 (embedding di un albero binario completo su una griglia bidimensionale)

Sia dato un albero binario completo T con n foglie e una griglia bidimensionale (foglio a quadretti). Si vuole disegnare T sulla griglia in modo da non far intrecciare gli archi e minimizzando lo spazio utilizzato. Più precisamente, un *embedding* di T è un disegno di T sul foglio tale che: (i) i nodi di T sono disegnati sulle intersezioni della griglia (ovvero come cerchi centrati su un qualche spigolo di un qualche quadratino), (ii) gli archi sono delle linee "seghettate" che seguono le linee del foglio, (iii) le linee che rappresentano gli archi non possono incrociarsi reciprocamente. Dato un embedding di T il suo *bounding box* è il rettangolo più piccolo che contiene l'intero embedding. Trovare un embedding di T che minimizza (asintoticamente) l'area del bounding box.

Esercizio 4 (Strutture dati: mantenere dinamicamente il mediano)

Si progetti una struttura dati che mantiene un insieme S di elementi con chiavi prese da un dominio totalmente ordinato soggetto alle seguenti operazioni:

- **Insert**(S, k): inserisce un nuovo elemento di valore k in S .
- **Delete**(S, x): rimuove l'elemento x da S . Si pensi ad x come al puntatore (riferimento diretto) all'elemento da cancellare.
- **Median**(S): restituisce (senza rimuoverlo) il valore dell'elemento mediano contenuto in S ; ovvero, se indichiamo con n il numero di elementi contenuti in S , si vuole restituire il valore dell' $\lceil \frac{n}{2} \rceil$ -esimo minimo di S .

Tutte le operazioni devono avere complessità temporale (asintoticamente) logaritmica nel numero di elementi presenti nella struttura dati.