

Esame del 19.09.2024

Algoritmi e Laboratorio

Parte B

Esercizio 1. Si consideri l'equazione di ricorrenza

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(\sqrt{n}). \quad (1)$$

Si risolva l'equazione (1) utilizzando il metodo preferito tra quelli studiati. Si stabilisca, inoltre, quali tra le seguenti condizioni sono soddisfatte dalla soluzione $T(n)$ all'equazione (1) trovata al punto precedente:

- | | |
|------------------------------|-------------------------|
| (a.) $T(n) = \mathcal{O}(n)$ | (b.) $T(n) = \Omega(n)$ |
| (c.) $T(n) = o(n)$ | (d.) $T(n) = \omega(n)$ |

Infine, si disegni uno sketch dell'albero di ricorrenza associato all'equazione (1), indicando il costo del livello i -esimo, l'altezza dell'albero e il numero di foglie.

Esercizio 2. Si consideri una tabella hash dove le collisioni sono risolte per concatenazione con m slot (o celle o caselle) in cui sono memorizzate n chiavi. Qual è il valore atteso del tempo computazionale necessario a una ricerca senza successo nella tabella? Si dimostri formalmente la correttezza della risposta alla domanda precedente.

Soluzioni

Esercizio 1. La funzione driving e la funzione watershed sono $f(n) = \Theta(\sqrt{n})$ e $w(n) = n^{\log_2 2} = n^1 = n$, rispettivamente. Perciò, per $0 < \varepsilon < \frac{1}{2}$ $f(n) = \Theta(\sqrt{n}) = \mathcal{O}(n^{\log_2 2 - \varepsilon})$. Applicando il Teorema Master otteniamo la soluzione $T(n) = \Theta(n)$.

$T(n) = \Theta(n)$ vuol dire che esistono $c_1, c_2 > 0$ e $n_0 \in \mathbb{N}$ tali che $c_1 n \leq T(n) \leq c_2 n$. La notazione big- \mathcal{O} si usa per i limiti superiori (ovvero $T(n) = \mathcal{O}(f(n))$ se esiste $c > 0$ e $n_1 \in \mathbb{N}$ tali che $T(n) \leq cf(n)$) e la notazione big- Ω si usa per i limiti inferiori (ovvero $T(n) = \mathcal{O}(f(n))$ se esiste $c > 0$ e $n_2 \in \mathbb{N}$ tali che $T(n) \geq cf(n)$) quindi **(a.) e (b.) sono vere**. La notazione little- o si usa per i limiti superiori non precisi (ovvero $T(n) = o(f(n))$ se per ogni $c > 0$ esiste $n_1 \in \mathbb{N}$ tali che $T(n) < cf(n)$) per $n \leq n_1$, quindi **(c.) è falsa**. Analogamente, la notazione little- ω si usa per i limiti inferiori non precisi quindi **(d.) è falsa**.

La radice ha costo $c\sqrt{n}$. Ogni nodo ha 2 figli. All' i -esimo livello dell'albero ci sono 2^i figli ognuno di costo $c\sqrt{\frac{n}{2^i}}$, per un costo del livello di $c\sqrt{2^i n}$. L'altezza dell'albero è $h = \log_2 n$ e ci sono n foglie.

Esercizio 2. Il valore atteso del tempo computazionale è $\mathcal{O}(1 + \alpha)$, dove $\alpha = \frac{n}{m}$ è il fattore di carico. Per dimostrare questo fatto, si osservi che il valore atteso per il tempo computazionale in una ricerca senza successo è il numero atteso di prove fatte, ovvero $T(n) = \Theta(1) + \mathbb{E}[X]$ dove $\Theta(1)$ indica il tempo impiegato per il calcolo del valore hash della chiave cercata e

$X = \{\text{numero di elementi da scannerizzare nella cella corrispondente all'hash della chiave}\}.$

Si ha $\mathbb{E}[X] = \alpha$ e pertanto $T(n) = \mathcal{O}(1 + \alpha)$.