Analisi – Ricorrenze

Si consideri la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} T(m) + T(n-m) + 1 & n > m \\ 1 & n \leq m \end{cases}$$

dove m è una costante intera positiva.

Si trovino, tramite il metodo della sostituzione, un limite superiore ed un limite inferiore per $T(n)$. Fare particolare attenzione ai casi base.

SortinoSort

Il professor Sortino ha inventato un nuovo algoritmo di ordinamento. Il vettore di input viene diviso in tre parti, di dimensioni approssimativamente uguali $n/3$. Dopo di che, vengono ordinati ricorsivamente le prime due parti del vettore (ovvero i primi due terzi), i secondi due terzi, e di nuovo i primi due terzi.

SortinoSort(**int**[ ] A, **int** i, **int** j)

```
if j − i + 1 ≤ 6 then
    InsertionSort(A, i, j)
else
    int s ← ⌈(j − i + 1)/3⌉
    SortinoSort(A, i, i + 2s − 1)
    SortinoSort(A, i + s, j)
    SortinoSort(A, i, i + 2s − 1)
```

SortinoSort

1. Qual è la complessità di questo algoritmo? Il Prof. Sortino finirà sulla prossima edizione del mio libro?
2. (Difficile, Opzionale) Dimostrare per induzione che questo algoritmo è corretto. Per comodità, assumete pure che tutti i valori siano distinti.

Alberi – Indovina l'albero

Gli ordini di visita di un albero binario di 9 nodi sono i seguenti:

- A, E, B, F, G, C, D, I, H (anticipato)
- B, G, C, F, E, H, I, D, A (posticipato)
- B, E, G, F, C, A, D, H, I (simmetrico).

Si ricostruisca l'albero binario e si illustri *brevemente* il ragionamento.

Alberi – Albero livello–valore

Scrivere un algoritmo che preso in input un albero binario T i cui nodi sono associati ad un valore intero $T.key$, restituisca il numero di nodi dell'albero il cui valore è pari al livello del nodo. Vi ricordo che il livello del nodo è pari al numero di archi che devono essere attraversati per raggiungere il nodo dalla radice. Per cui la radice ha livello 0, i suoi figli hanno livello 1, etc.

Alberi – Cammino radice–discendente crescente

Dato un albero binario contenente interi, scrivere un algoritmo che restituisca la lunghezza del più lungo cammino monotono crescente radice-discendente, dove:

- il discendente non è necessariamente foglia;
- con lunghezza si intende il numero totale di *archi* attraversati;
- con monotona crescente si intende che i valori contenuti nei nodi della sequenza devono essere ordinati in senso crescente da radice a discendente.

Discuterne correttezza e complessità.

Spoiler alert!

Ricorrenza $T(n) = T(m) - T(n-m) + 1$

- m è un valore costante, quindi è costante;
- Il numero di chiamate ricorsive è $\lfloor n/m \rfloor$;
- Ogni chiamata costa $T(m) + 1$, ancora un valore costante.

Supponiamo quindi che $T(n) = \Theta(n)$.

Ricorrenza $T(n) = T(m) - T(n-m) + 1 = O(n)$

Passo induttivo: supponiamo per ipotesi induttiva che $T(n') \leq cn'$ per ogni $n' < n$. Proviamo che $\exists c > 0 : T(n) \leq cn$.

$$\begin{aligned} T(n) &= T(m) + T(n-m) + 1 \\ &\leq 1 + cn - cm + 1 \\ &\leq cn - cm + 2 \\ &\overset{?}{\leq} cn \end{aligned}$$

La disequazione è vera per $c \geq 2/m$; poichè $m \geq 1$, $c \geq 2$ è un valore accettabile per ogni m.

Ricorrenza $T(n) = T(m) - T(n-m) + 1 = O(n)$

Caso base: consideriamo tutti i casi per cui $1 \leq i \leq m$:

$$T(i) = 1 \leq c \cdot i \Leftrightarrow c \geq \frac{1}{i}$$

Le disequazioni su c derivanti dal caso base possono essere riassunte dal fatto che $c \geq 1$, poichè $1/i \leq 1$ per tutti i valori $i \geq 1$.

Considerando quindi sia la disequazione sul passo ricorsivo che sul caso base, otteniamo che $c \geq 2$.

Ricorrenza $T(n) = T(m) - T(n-m) + 1 = \Omega(n)$

Passo induttivo: supponiamo per ipotesi induttiva che $T(n') \geq cn'$ per ogni $n' < n$. Proviamo che $\exists c > 0 : T(n) \geq cn$.

$$\begin{aligned} T(n) &= T(m) + T(n-m) + 1 \\ &\geq 1 + cn - cm + 1 \\ &\geq cn \end{aligned}$$

L'ultima disequazione è banalmente vera per ogni $c \leq 2/m$.

Caso base: consideriamo tutti i casi per cui $1 \leq i \leq m-1$:

$$T(i) = 1 \geq c \cdot i \Leftrightarrow c \leq \frac{1}{i}$$

Il valore $\frac{1}{m}$ è il più piccolo fra tutte le disequazioni trovate; per soddisfarle tutte, compresa quella derivante dal passo induttivo, basterà porre $c = 1/m$.

SortinoSort – 12/09/10

La complessità è rappresentata dalla seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 3T(\frac{2}{3}n) + 1 & n > 6 \\ 1 & n \leq 6 \end{cases}$$

Utilizzando il teorema delle ricorrenze lineari con partizione bilanciata, si ottiene che $a = 3$, $b = 3/2$, da cui $\alpha = \log_{3/2} 3$; inoltre, $\beta = 0$. Siamo quindi nel caso $T(n) = n^{\alpha}$.
Non avete una calcolatrice e non sapete quanto sia $\log_{3/2} 3$?

SortinoSort – 12/09/10

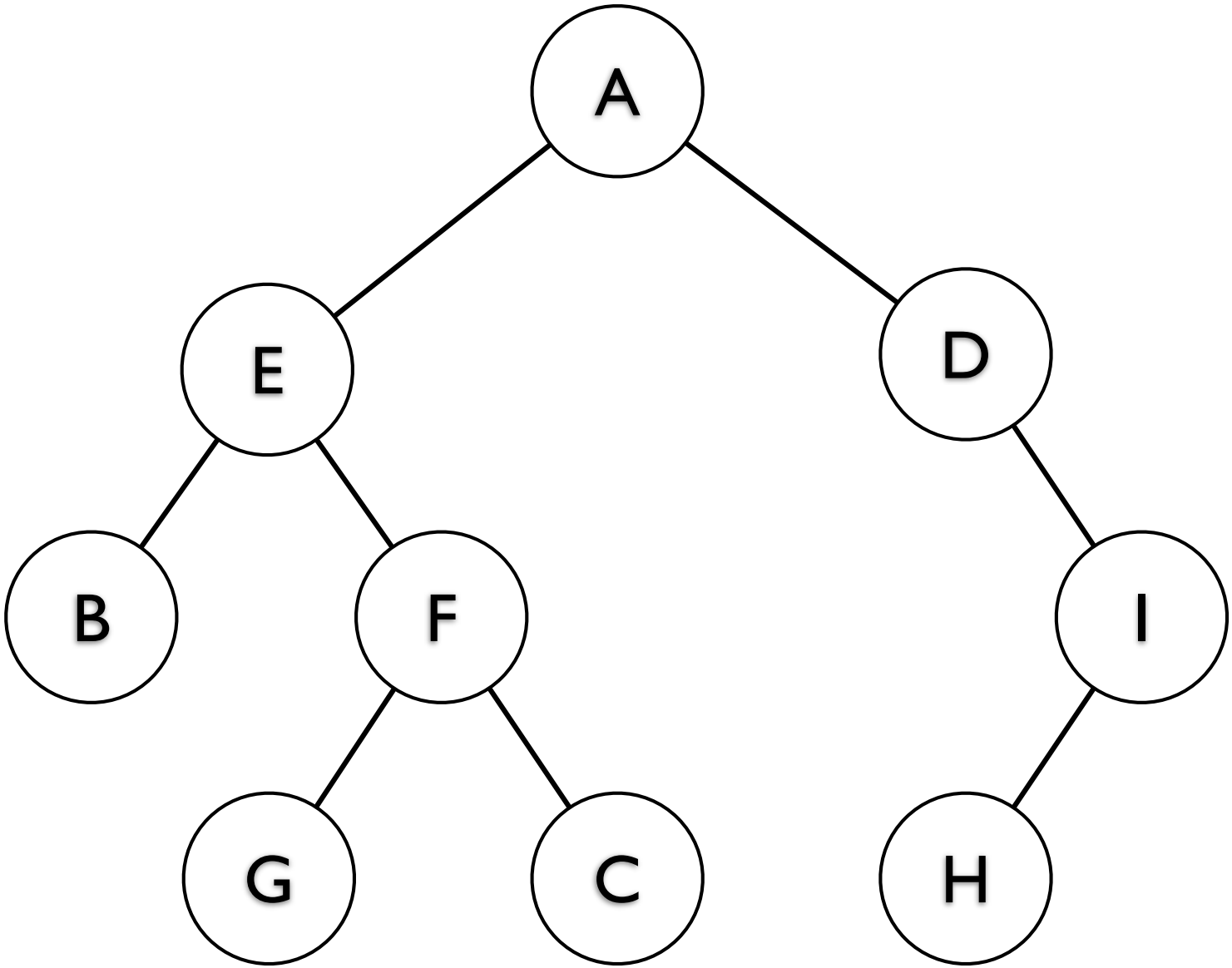
La complessità è rappresentata dalla seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 3T(\frac{2}{3}n) + 1 & n > 6 \\ 1 & n \leq 6 \end{cases}$$

Utilizzando il teorema delle ricorrenze lineari con partizione bilanciata, si ottiene che $a = 3$, $b = 3/2$, da cui $\alpha = \log_{3/2} 3$; inoltre, $\beta = 0$. Siamo quindi nel caso $T(n) = n^{\alpha}$.
Non avete una calcolatrice e non sapete quanto sia $\log_{3/2} 3$?

E' semplice: $\left(\frac{3}{2}\right)^2 = \frac{9}{4} < 3$, mentre $\left(\frac{3}{2}\right)^3 = \frac{27}{8} > 3$. Quindi α è compreso fra 2 e 3, e quindi questo algoritmo è addirittura peggiore di Insertion Sort. Il prof. Sortino non finirà nel mio libro.

Indovina l'albero



Albero livello-valore

Una semplice visita posticipata risolve il problema. La complessità è ovviamente $O(n)$.

---

int sameLevel(TREE T)

---

return sameLevelRec(T, 0)

---

int sameLevelRec(TREE T, **int** ℓ)

---

int $tot = 0$
if $T \neq$ **nil then**
  $tot =$ sameLevelRec(T.right(), $\ell + 1$) + sameLevelRec(T.left(), $\ell + 1$)
  if $T.key == \ell$ **then**
    $tot = tot + 1$
return tot

Alberi – Percorso cammino-discendente

int monotone(TREE T)

int $maxl = 0$
int $maxr = 0$
if $T \neq$ **nil then**
    if T.left() $\neq Nil$ **and** T.left().$value > T.value$ **then**
        $maxl = 1 +$ monotone(T.left())
    if T.right() $\neq Nil$ **and** T.right().$value > T.value$ **then**
        $maxr = 1 +$ monotone(T.right())
return max($maxl, maxr$)