Algoritmi e Strutture Dati
09/07/12

Esercizio 1

Poiché m è un valore costante, ovviamente anche $T(m)$ lo è. È facile notare che il numero di chiamate ricorsive sarà pari a $\lfloor n/m \rfloor$. Ognuna di queste chiamate costerà $T(m) + 1$, quindi ancora un valore costante. Possiamo quindi suppore un limite superiore e inferiore $\Theta(n)$.

$O(n) \quad :$
Come passo induttivo, supponiamo di aver provato per tutti i casi $n' < n$ che $T(n') \leq cn'$; vogliamo ora provare che $\exists c > 0 : T(n) \leq cn$.

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

La disequazione è vera per $c \geq 2/m$; poichè $m \geq 1$, $c \geq 2$ è un valore accettabile per ogni m.
Come case base, consideriamo tutti i casi per cui $1 \leq n \leq m$:

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

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 $c \geq 2$.

$\Omega(n) \quad :$
Come passo induttivo, supponiamo di aver provato per tutti i casi $n' < n$ che $T(n') \geq cn'$; vogliamo ora provare 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 $c \leq 2/m$.
Come 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 $c = 1/m$.

Esercizio 2

Tramite gli algoritmi che risolvono il problema della selezione visti a lezione, possiamo individuare la mediana in tempo (atteso) $O(n)$. Una volta individuata la mediana, possiamo costruire un vettore di appoggio che contenga la differenza fra gli elementi di A e la mediana. Utilizzando di nuovo l'algoritmo della selezione, possiamo individuare la k-esima differenza rispetto alla mediana. Utilizzando questo valore, possiamo individuare tutti i valori la cui distanza è inferiore.

```
kmediana(int[] A, int n, int k)
    int m = selezione(A, ⌊n/2⌋)
    int[] B = new int[1...n]
    for i = 1 to n do
        B[i] = |A[i] − m|
    int d = selezione(B, k)
    for i = 1 to n do
        if |A[i] − m| ≤ d then
            print A[i]
```