Algoritmi e Strutture Dati
Soluzione di esercizi proposti
(Prima settimana)

Complessità asintotica esatta θ

Dimostrare che

$3/2\, n^2 - 2n$ è $\theta(n^2)$

bisogna dimostrare che

esistono costanti positive n_0, c_1, c_2 tali che

$c_1 n^2 \le 3/2\, n^2 - 2n \le c_2 n^2$, per ogni $n > n_0$

divido per n^2

$c_1 \le 3/2 - 2/n \le c_2$

- $3/2 - 2/n \le c_2$

Per n=2 la parte sinistra della disuguaglianza è positiva, al crescere di n inoltre, tende a 3/2. Quindi è maggiorata da 3/2, che è il valore scelto per c_2.

Esiste $n_0 = 2$ e $c_2 = 3/2$ tali che, per ogni $n > 2$, $3/2 - 2/n \le c_2$

… continua

- $c_1 <= 3/2 - 2/n$

$n_1=2$ è il più piccolo intero positivo che rende $3/2 - 2/n$ positiva. Per $n=2$, $3/2 - 2/n = 1/2$.

$c_1=1/2$ sarà sempre minore di $3/2-2n$ (per $n>=2$)

Esiste $n_1= 2$ e $c_1 = 1/2$ tali che, per ogni $n>1/2$,

$c_1 <= 3/2 - 2/n\ c_2$

Scelto $n'=max(n_0, n_1)$ (in questo caso sono uguali) il teorema è stato dimostrato:

esistono costanti positive $n'=2$, $c_1=1/2$, $c_2=3/2$ tali che

$c_1 n^2 <= 3/2\ n^2 - 2n <= c_2 n^2$, *per ogni* $n > n'$

Ricerca Binaria

- **Dato un array di *n* interi, ordinati in ordine crescente e dato un intero *key*, restituire la posizione dell'inidice dell'array in cui si trova *key*; se *key* non è presente nell'array, restituire il valore *0*.**

- **Scrivere la pseudocodifica dell'algoritmo usando la tecnica della ricerca binaria;**
- **Calcolare la complessità asintotica dell'algoritmo.**

pseudocodifica

```
1. BIN-SEARCH(A,min,max,key)
2. if min>max
3.    then return 0
4.    else med <-(min+max)/2
5.         if key=A[med]
6.            then return med
7.            else if key<A[med]
8.                 then BIN-SEARCH(A,min,med-1,key)
9.                 else BIN-SEARCH(A,med+1,max,key)
```

- Le condizioni alle righe 2 e 5 corrispondono al passo base della ricorsione
- Quando min>max la dimensione dell’array è 0 (gli indici si sono incrociati) e il costo è costante: c_0

- Le operazioni alle righe 4-7 sono a costo costante: c_1
- L’algoritmo viene richiamato su un input di dimensioni dimezzate

L’algoritmo verrà richiamato dal main progam con

`BIN-SEARCH(A,1,length[A],k)`

Complessità asintotica

$$T_{BIN\_SEARCH}(n) = \begin{cases} c_0 & \text{per } n=0 \\ T_{BIN\_SEARCH}(n/2) + c_1 & \text{per } n>0 \end{cases}$$

Per il Master Theorem *a=1, b=2, k=0*
$a=b^k$ (secondo caso)
che fornisce una complessità $\theta(n^k \log n)$
quindi si ha $\theta(\log n)$