In questa lezione:

Esempi di analisi di algoritmi ricorsivi

Ricerca binaria: versione ricorsiva

INPUT: una sequenza A di n numeri e l'elemento da cercare, *x*.
PREC: gli elementi di A sono <u>ordinati in ordine crescente</u>
OUTPUT: la posizione di *x* se occorre nella sequenza, -1 altrimenti.

RicBinariaRic(A,x)
sia n il numero degli elementi di A
<u>if</u> n = 0 <u>then return</u> -1
<u>if</u> x == A[n/2] <u>then return</u> n/2
<u>else if</u> (x < A[n/2])

$\Theta(1)$

<u>then</u> chiama la funzione sugli elementi $\leq T(n/2)$
nella metà sinistra di A (escluso A[n/2])
<u>else</u> chiama la funzione sugli elementi $\leq T(n/2)$
nella metà destra di A (escluso A[n/2])

Implementazione Python:
http://interactivepython.org/runestone/static/pythonds/SortSearch/TheBinarySearch.html#analysis-of-binary-search

La ricorrenza

$T(n) = T(n/2) + \Theta(1)$

Esplicitiamo le costanti e il caso base:

$T(n) = T(n/2) + c$

$T(1) = d$

Ipotizziamo che $T(n)$ sia in $\Theta(\lg n)$

O(lg n) per Ricerca Binaria

Dimostriamo che $T(n) = O(\lg n)$.

<u>Ipotesi induttiva</u>: supponiamo che $T(m) \leq k \lg m$ per ogni $m < n$
e consideriamo $T(n)$:

$$T(n) \leq T(n/2) + c$$

$\leq k \lg n/2 + c$ **perché** $T(n/2) \leq k \lg n/2$ **(ip. ind.)**

$\leq k (\lg n - \lg 2) + c = k \lg n - k + c$

$\leq k \lg n$ **prendendo** $k \geq c$ **e** $n_0 = 2$,
perché $\lg 1 = 0$

Concludiamo che $T(n) = O(\lg n)$.

$\Omega(\lg n)$ per Ricerca Binaria

Dalla relazione il limite inferiore banale:
$T(n) = \Omega(1)$

Dimostriamo che $T(n) = \Omega(\lg n)$.
Supponiamo che $T(m) \geq k \lg m$ per ogni $m < n$ e consideriamo $T(n)$:

$$
\begin{aligned}
T(n) &= T(n/2) + c \\
&\geq k \lg n/2 + c \\
&\geq k \lg n - k + c \\
&\geq k \lg n \text{ per ogni } k \leq c
\end{aligned}
$$

Concludiamo che $T(n) = \Theta(\lg n)$.

Esempio algoritmo ricorsivo

```
analizzami(int n)
c = 1
m = n                      Θ(1)
while (m > 1) :
    c++                    Θ(lg n)
    m=m/2
if n >1 then analizzami(n/2)
```

$$T(n) = T(n/2) + \Theta(\lg n)$$

Soluzione

$$T(n) = \begin{cases} \Theta(1) & \text{se } n = 1 \\ T(n/2) + c\lg n & \text{se } n > 1 \end{cases}$$

Se $2^h \leq n < 2^{h+1}$ la parte intera della divisione di n per 2^h è 1, e h è la parte intera di lg n

$$T(n) = T(n/2) + c\lg n$$

$$= T(n/2^2) + c\lg n/2 + c\lg n$$

$$= T(n/2^3) + c\lg n/2^2 + c\lg n/2 + c\lg n$$

$$\ldots$$

$$= T(1) + \sum_{i=0,\ldots,h-1} c\lg n/2^i$$

$$= T(1) + c\sum_{i=0,\ldots,h-1} (\lg n - i)$$

$$= T(1) + (h-1)\lg n - h(h-1)/2 = \Theta(\lg^2 n)$$

$O(\lg^2 n)$

Dimostriamo che $T(n) = O(\lg^2 n)$.

Ipotesi induttiva: **supponiamo che $T(m) \leq k \lg^2 m$ per ogni $m < n$**
e consideriamo $T(n)$:

$$T(n) = T(n/2) + c \lg n \leq k \lg^2 n/2 + c \lg n$$

$$\leq k (\lg n - 1)^2 + c \lg n$$

$$\leq k (\lg^2 n + 1 - 2 \lg n) + c \lg n =$$

$$k \lg^2 n + k - 2 k \lg n + c \lg n$$

Si vuole che

$k \lg^2 n \underbrace{+ k - 2 k \lg n + c \lg n} \leq k \lg^2 n$, cioè che

$k - 2 k \lg n + c \lg n \leq 0$, cosa vera per esempio per $k = c$ e $n \geq 2$

$\Omega(\lg^2 n)$

Dimostriamo che $T(n) = \Omega(\lg^2 n)$.

Ipotesi induttiva: **supponiamo che $T(m) \geq k \lg^2 m$ per ogni $m < n$**
e consideriamo $T(n)$:

$$T(n) = T(n/2) + c \lg n \geq k \lg^2 n/2 + c \lg n \geq$$
$$k (\lg n - 1)^2 + c \lg n \geq k(\lg^2 n + 1 - 2\lg n) + c\lg n$$
$$= k\lg^2 n + k - 2k\lg n + c \lg n$$

Si vuole che
$k \lg^2 n + \underbrace{k - 2 k \lg n + c \lg n} \geq k \lg^2 n$, cioè che

$k - 2 k \lg n + c \lg n \geq 0$,
cosa vera per esempio per $k = c/2$ e $n \geq 1$