Titolo nota 17/03/2016

A di n interi distinti

$A[i]$ $A[j]$ $i \neq j$:

$$|A[i] - A[j]| \leq |A[k] - A[l]|$$

$$\sum_{i=1}^{n-1} (n-i) = \sum_{j=1}^{n-1} j = \frac{n(n-1)}{2} = \Theta(n^2)$$

$j = n - i = n - n + 1 = 1$

$i = n-1$

<u>Soluzione banale</u> : si analizzano tutte le possibili coppie e si determina quella a distanza minima

```
imin = 0
min = ∞
for (i = 1; i < n; i++)
  for (j = i+1; j ≤ n; j++) {
      if |A[i] - A[j]| < min
         min = |A[i] - A[j]|;
         imin = i;
  }
```

$\Theta(n^2)$ $\Theta(n^2)$ $\Theta(1)$

$$\binom{n}{2} = \frac{n!}{2(n-2)!} = \frac{n(n-1)\cancel{(n-2)}\cancel{\cdots}\cancel{1}}{2\cancel{(n-2)}\cancel{(n-3)}\cdots\cancel{1}} = \frac{n(n-1)}{2} = \Theta(n^2)$$

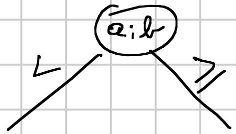
MergeSort $(A, 1, n)$;

$-5 \ -1 \ 2 \ 6 \ 8 \ 10 \ 11 \ 17 \ 18$ $\Rightarrow$ la coppia che minimizza la dist. è costituita da el. consecutivi nell'array ordinato.

```
imin = 1; min |A[1] - A[2]|;
for (i = 2; i < n; i++) {
  if |A[i] - A[i+1]| < min {
      min = |A[i] - A[i+1]|;
      imin = i;
} return imin;
```

$\Theta(n \log n) + \Theta(n) = \Theta(n \log n)$

Limite Inferiore: $\Omega(n)$ con la dimensione dell'input.



selezionare una coppia tra tutte le possibili che sono in numero:

$$|S| = \binom{n}{2} = \frac{n(n-1)}{2}$$

$$c \geq \log|S| \geq \Theta(\log n^2) \geq \Theta(2\log n) = \Theta(\log n)$$

è $\Omega(\log n)$ è significativo? NO

Titolo nota 17/03/2016

1) $T(n) = 8\,T\left(\frac{n}{2}\right) + n$

2) $T'(n) = a\,T'\left(\frac{n}{4}\right) + n$ a costante

det. il num. a : $T'(n)$ cresce più di $T(n)$

$a < 4$

$a = 4$

$4 < a < 64$

$a > 64$

1) $a = 8$ $b = 2$ $n^{\log_2 8} = n^3$

$n : n^3$ $n = O(n^{3-\varepsilon})$ vero per $\varepsilon \leq 2$ $T(n) = \Theta(n^3)$

2) $a < 4$ $b = 4$ $n^{\log_4 a}$

$n = \Omega\left(n^{\log_4 a + \varepsilon}\right)$ se $\log_4 a + \varepsilon \leq 1$, $\varepsilon \leq 1 - \log_4 a$

$\boxed{T'(n) = n}$ se vale $a\,f\left(\frac{n}{b}\right) \leq c\,f(n)$ con $c < 1$

$a\,\frac{n}{4} \leq cn$ $a\,\frac{n}{4} \leq cn$ vale c $c \geq \frac{a}{4}$ caso 3

Titolo nota 17/03/2016

$a = 4$ $\qquad n^{\log_{64} 4} = f(n) = n$

caso 2 $\qquad T'(n) = \Theta(n \log n)$

$\boxed{a = 64}$ $\qquad n^{\log_{64} 64} = n^3$

$n = O(n^{3-\varepsilon})$

caso 1 $\qquad T'(n) = \Theta(n^3)$

$a > 64$ $\qquad T'(n)$ supera $T(n)$ $\qquad$ min $a = 65$

ES. 4 La coppia che massimizza la distanza.

la risposta è $|MAX - min|$

Determina min e Max di array A con algoritmo tipo "torneo"

MinMax (A, sx, dx)   1ª chiamata sx = 1   dx = n

Risolvi dir. { if (sx == dx) return <A[sx], A[sx]> restituisce una coppia <min, max>

else {

Dividi: centro = (sx + dx)/2;

Risolvi Ricorsivamente {
<m_1, M_1> = MinMax (A, sx, centro);
<m_2, M_2> = MinMax (A, centro+1, dx);

$$\begin{cases} T(n) = \Theta(1) & \text{se } n = 1 \\ T(n) = 2T\left(\frac{n}{2}\right) + \Theta(1) & \text{se } n > 1 \end{cases}$$

Combina $\Theta(1)$ {
if ($m1 < m_2$) min = m_1;
else min = m2;

if (M1 > M2) max = M1;
else MAX = M2;

$$T(n) = \Theta(n)$$

return <min, MAX>;

A n elementi 0, 1 più 0 che 1 ?

Restituisce diff = #0 − #1

se diff = 0 #0 = #1

se diff > 0 #0 > #1

se diff < 0 #0 < #1

0	1	1	0	1	1	1	0
1	−1	−1	1	−1	−1	−1	1
0		0		−2		0	
0				−2			
−2							

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

$$T(n) = \Theta(n)$$

```
DIFF01(A, sx, dx):
if(sx==dx){
    if A[sx]==0 return 1;
    else return -1;
} else { cx = (sx+dx)/2;
    diff1= DIFF01(A, sx, cx);
    diff2 = DIFF01(A, cx+1, dx);
    diff= diff1+diff2;
    return diff;
}
```

$\Theta(1) \rightarrow$

Correttezza Induzione su n

base : $n = 1$ vero si risolve per ispezione diretta

i : posto vero $n' < n \Rightarrow n$

$$diff = diff_1 + diff_2$$

i.i $\Rightarrow$

$$= \underbrace{\#0_1 - \#1_1} + \underbrace{\#0_2 - \#1_2}$$

$$= \underbrace{\#0_1 + \#0_2} - \underbrace{(\#1_1 + \#1_2)}$$

$$= \#0 - \#1$$

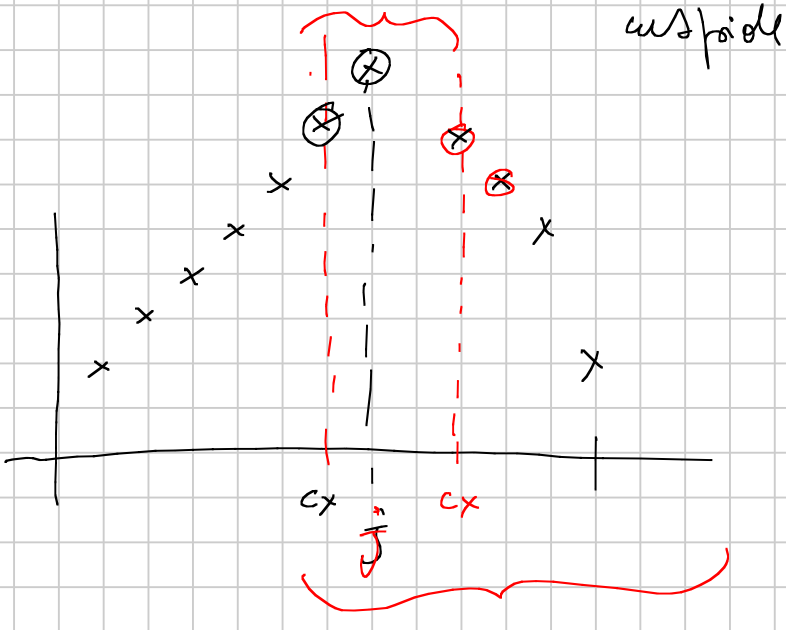
Titolo nota 17/03/2016

A n elementi positivi

$A[1..j]$ è crescente

$A[j+1,..n]$ è descrescente

$A[j] > A[j+1]$ $j < n$



Algoritmo banale: $O(n)$

```
TROVA POS (A, sx, dx):
if (sx == dx) return sx;
  else {
  cx = (sx + dx)/2;
  if (A[cx] < A[cx+1]) return TrovaPos(A, cx+1, dx);
     else return TrovaPos(A, sx, cx);
  }
```

$$\begin{cases} T(n) = \Theta(1) & n=1 \\ T(n) = T\left(\frac{n}{2}\right) + \Theta(1) \end{cases}$$

$$T(n) = \Theta(\log n)$$

$C \geq \log |S|$

a : b

< ≥

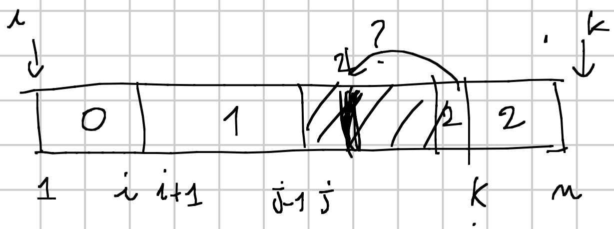
albero di decisione binario

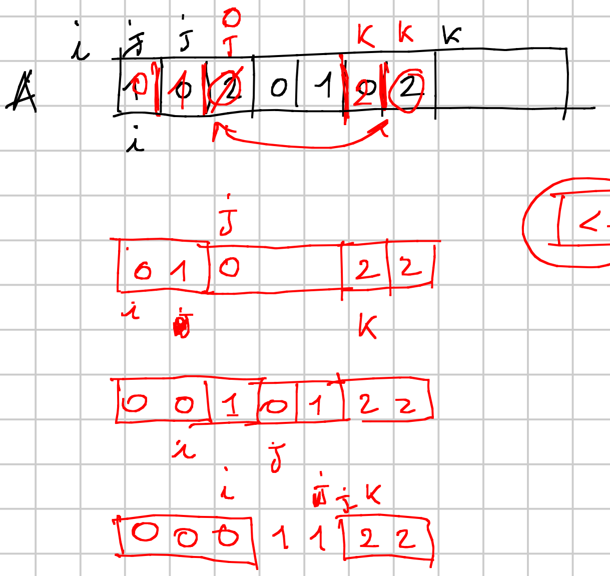
$C \geq \log_2 n+1$ $\qquad |S|$? $\qquad \Rightarrow \Omega(\log n)$

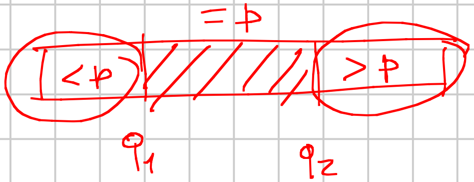
alg. ottimo

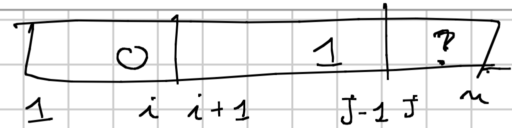
Titolo nota 17/03/2016

Ordina A n interi 0, 1, 2









ORDINA012 (A):

```
i = 0;  K = n+1;  j = 1;
while ( j < K ) {
    if ( A[j] == 0 ) {
        i++;  scambia A[i] e A[j];
        j++;
    } if ( A[j] == 2 ) {
        K--;  scambia A[j] e A[K];
    } else  // A[j] = 1;
        j++;
}
```

non si incrementa j

Titolo nota 17/03/2016

Ciclicamente ordinato

j

$a = 12, 14, 20, 24, 1, 3, 7, 11$

Puzzle cavalli

25 cavalli corse a 5 cavalli ciascuna

5 corse

un ultima corsa con i vincitori per stabilire il campione } 6 corse ALG.

Titolo nota 17/03/2016

6 corse sono necessarie

1) Tutti devono partecipare ad almeno una corsa per essere classificati

2) Per poter essere NON CAMPIONE un cavallo deve aver perso in almeno una corsa

3) Con 5 corse quanti non campioni 20

4) Quanti devono essere i non campioni 24.
Occorrono almeno 6 corse