```
SommaMassima1( a ):    ⟨pre: a contiene n elementi di cui almeno uno positivo⟩
  max = 0;
  FOR (i = 0; i < n; i = i+1) {
    FOR (j = i; j < n; j = j+1) {
      somma = 0;
      FOR (k = i; k <= j; k = k+1)
        somma = somma + a[k];
      IF (somma > max) max = somma;
    }
  }
  RETURN max;
```

$O(1)$ $O(1)$ $O(1)$ $O(1)$

$\Theta(j-i+1)$

$\Theta(n^3)$

$n-i-1$

$n-i$ $\times$

$\Theta(n-i) + \sum_{j=i} \Theta(j-i+1)$

$k=0$

$\Theta((n-i)^2)$

$t_j = \Theta(j-i+1)$

```
SommaMassima1( a ):   ⟨pre: a contiene n elementi di cui almeno uno positivo⟩
  max = 0;
  FOR (i = 0; i < n; i = i+1) {
    FOR (j = i; j < n; j = j+1) {
      somma = 0;
      FOR (k = i; k <= j; k = k+1)
        somma = somma + a[k];
      IF (somma > max) max = somma;
    }
  }
  RETURN max;
```

$\Theta((n-i)^2)$

$t_i = \Theta((n-i)^2)$

$\Theta(n) + \sum_{i=0}^{n-1} \Theta((n-i)^2) = \Theta(n^3)$

$$\sum_{i=0}^{n-1} \Theta((n-i)^2) = \Theta(n^3)$$

$\vee$

$$= O(n^3)$$ $n-i \le n$

$$\sum_{i=\frac{n}{4}}^{n/2} \Theta((n-i)^2) \geq \sum_{i=\frac{n}{4}}^{n/2} \Theta\Big(\underbrace{\big(n-\tfrac{n}{2}\big)^2}_{\frac{n}{2}}\Big) = \Theta\left(\frac{n^2}{4}\right)$$

$\Theta(n^2)$

$$\geq \Omega\left(\frac{n^3}{8}\right) = \Omega(n^3)$$

```
SommaMassima2( a ): ⟨pre: a contiene n elementi di cui almeno uno positivo⟩
  max = 0;
  FOR (i = 0; i < n; i = i+1) {
    somma = 0;
    FOR (j = i; j < n; j = j+1) {
      somma = somma + a[j];
      IF (somma > max) max = somma;
    }
  }
  RETURN max;
```

$O(1)$

$O(1)$

$O(1)$

$\Theta(n-i)$

$$\Theta(n) + \sum_{i=0}^{n-1} \Theta(n-i) = \Theta\left(\sum_{i=0}^{n-1} \underbrace{(n-i)}_{\ell}\right) = \Theta\left(\sum_{\ell=1}^{n} \ell\right) = \Theta(n^2)$$

$A[i,j]$ è massimale se non esiste $A[i',j']$ t.c. $i' \leq i \leq j \leq j'$ e

$$\text{somma}(i',j') > \text{somma}(i,j)$$

2	-4	1	3	-7	-2	-1	2	-1	3	-2	-1	5	...

$i = j'$

j

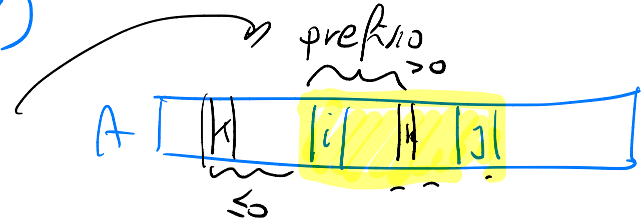
```
SommaMassima3( a ):    ⟨pre: a contiene n elementi di cui almeno uno positivo⟩
  max = 0;
  somma = max;
  FOR (j = 0; j < n; j = j+1) {
O(1) IF (somma > 0) {
O(1)   somma = somma + a[j];
O(1) } ELSE {
O(1)   somma = a[j];        i = j
     }
O(1) IF (somma > max) max = somma;
  }
  RETURN max;
```

posso ricordarmi il segmento i, j t.c. somma(i,j) = max

$$\sum_{j=0}^{n-1} O(1) = O(n)$$

Sia $A[i,j)$ un segmento di somma massimale (massimale?)

(a) Ogni prefisso di $A[i,j)$ ha somma positiva: $\text{somma}[i,k) > 0 \ \forall k : i \le k < j$



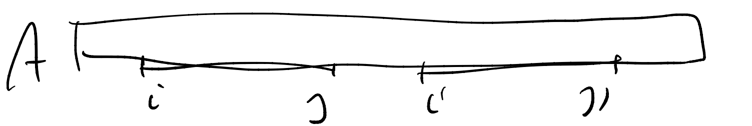
(b) $A[i,j]$ non può essere esteso a sinistra

$\text{somma}(k, i-1) \le 0 \quad \forall k < i$

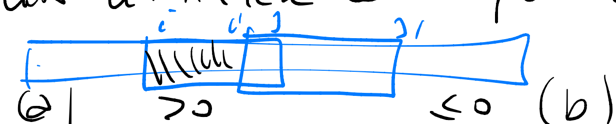
Proprietà

Siano $A[i,j)$ e $A[i',j')$ due segmenti che soddisfano (a) e (b).

Allora $A[i,j) \cap A[i',j') = \emptyset$



Suggerimento: utilizzare dim. per assurdo



SI PUÒ FARE DI MEGLIO ?

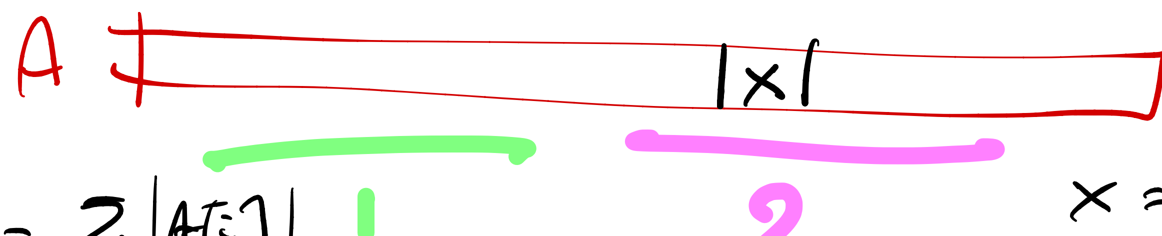
LIMITE INFERIORE (LOWER BOUND)

<u>NO</u> : Leggendo soltanto $\leq$ <u>n-1</u> elementi in A non possiamo sempre dare risposte corrette

FOOLISH ARGUMENT

$\Omega(n)$

non è letto



$x = \sum_i |A[i]|$

$x = -\sum_i |A[i]|$

COMPLESSITÀ

- Algoritmo A: complessità asintotica = ordine di grandezza del corrispondente numero di passi in funzione di n (dim. input)

  - caso pessimo max
  - caso medio $\frac{\Sigma}{\#\text{casi}}$

$O(f(n))$ limite superiore

Problema computazionale Π

tempo
spazio

- LIMITE SUPERIORE $O(f(n))$

  se esiste un algoritmo A per Π
  con complessità $O(f(n))$

- LIMITE INFERIORE $\Omega(g(n))$

  ogni algoritmo A per Π
  ha una complessità che
  non può scendere asintoticamente
  sotto $g(n)$

$\rightarrow \Theta(f(g))$

$\Theta(f(n)) = \Theta(g(n))$

<u>Esempio</u> A, B matrici $m \times m$

$$(A \times B)_{ij} = \sum_{k=1}^{m} A_{ik} \cdot B_{kj} \qquad O(m)$$

$$O(m^3) = O(n^{3/2})$$

$n = \text{dim. input} = m^2$

$$\Omega(n) = \Omega(m^2) \text{ per leggere}$$

Complessità? $O(m^{2.3728596})$

$O(m^{2+\varepsilon})$ $0 < \varepsilon < 1$

ORDINAMENTO (SORTING)

INPUT Array A di n elementi presi da un dominio con relazione d'ordine $<$

OUTPUT Riorganizzazione di A t.c.

$$A[0] < A[1] < \dots < A[n-1]$$

LIMITE INFERIORE se si contano i confronti $<$

$\Omega(n \lg n)$

UPPER BOUND

$O(n^2)$, $O(n \lg n)$

$\rightarrow$ complessità dell'ordinamento mediante confronti è $\Theta(n \lg n)$