Sottoarray di somma massima

Sezione 2.1 [CGG]

Redirezione dell'input

Meccanismo per prendere dati in input direttamente da file di testo. Questa tecnica sarà utilizzata per la correzione automatica delle prove di laboratorio.

Sintassi Tipica: prog < input

> ./esercizio.o < input (Linux)
> esercizio.exe < input (Windows)

Importante: durante le prove di laboratorio attenersi strettamente alle specifiche sul formato dell'output.

Sotto Array di Somma Massima

Problema: data una sequenza di interi (anche negativi) individuare la sottosequenza di somma massima.
Input: un array di interi (anche negativi)
Output: valore della somma

Esempio:

Input -1 5 8 -9 1 1

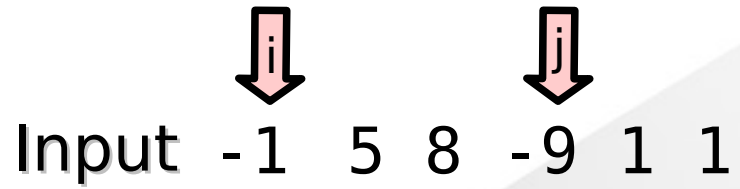
Output 13

Soluzione 1

```
max=a[0];
for(i=0; i<n; i++)
{
   for(j=i; j<n; j++)
   {
      somma=0;
      for(k=i; k<=j; k++)
      {
         somma+=a[k];
      }
      if(somma > max) max=somma;
   }
}
```

Soluzione 1

```
max=a[0];
for(i=0; i<n; i++)
{
   for(j=i; j<n; j++)
   {
      somma=0;
      for(k=i; k<=j; k++)
      {
         somma+=a[k];
      }
      if(somma > max) max=somma;
   }
}
```



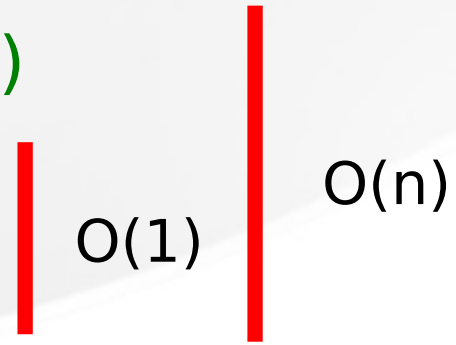
Soluzione 1

```
max=a[0];
for(i=0; i<n; i++)
{
   for(j=i; j<n; j++)
   {
      somma=0;
      for(k=i; k<=j; k++)
      {
         somma+=a[k];
      }
      if(somma > max) max=somma;
   }
}
```

O(1)

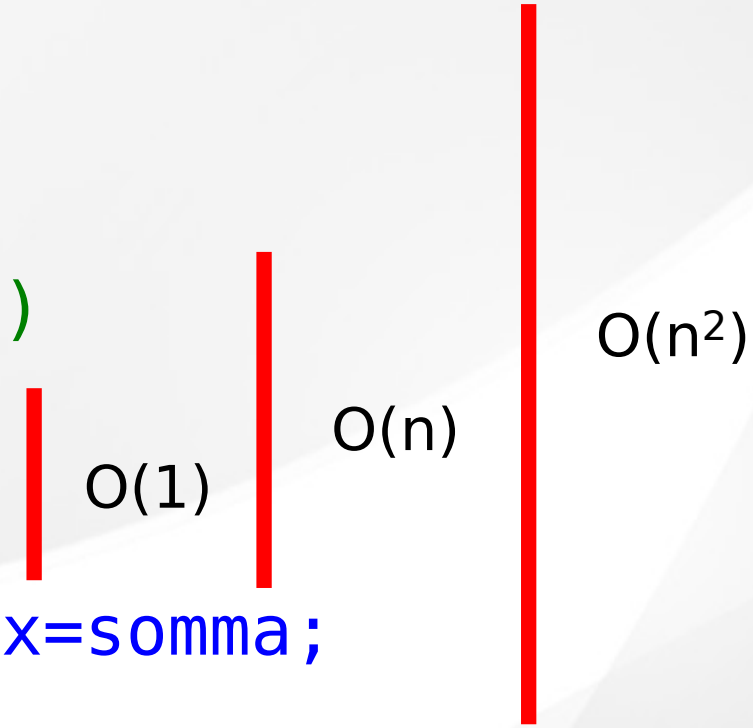
Soluzione 1

```
max=a[0];
for(i=0; i<n; i++)
{
   for(j=i; j<n; j++)
   {
      somma=0;
      for(k=i; k<=j; k++)
      {
         somma+=a[k];
      }
      if(somma > max) max=somma;
   }
}
```



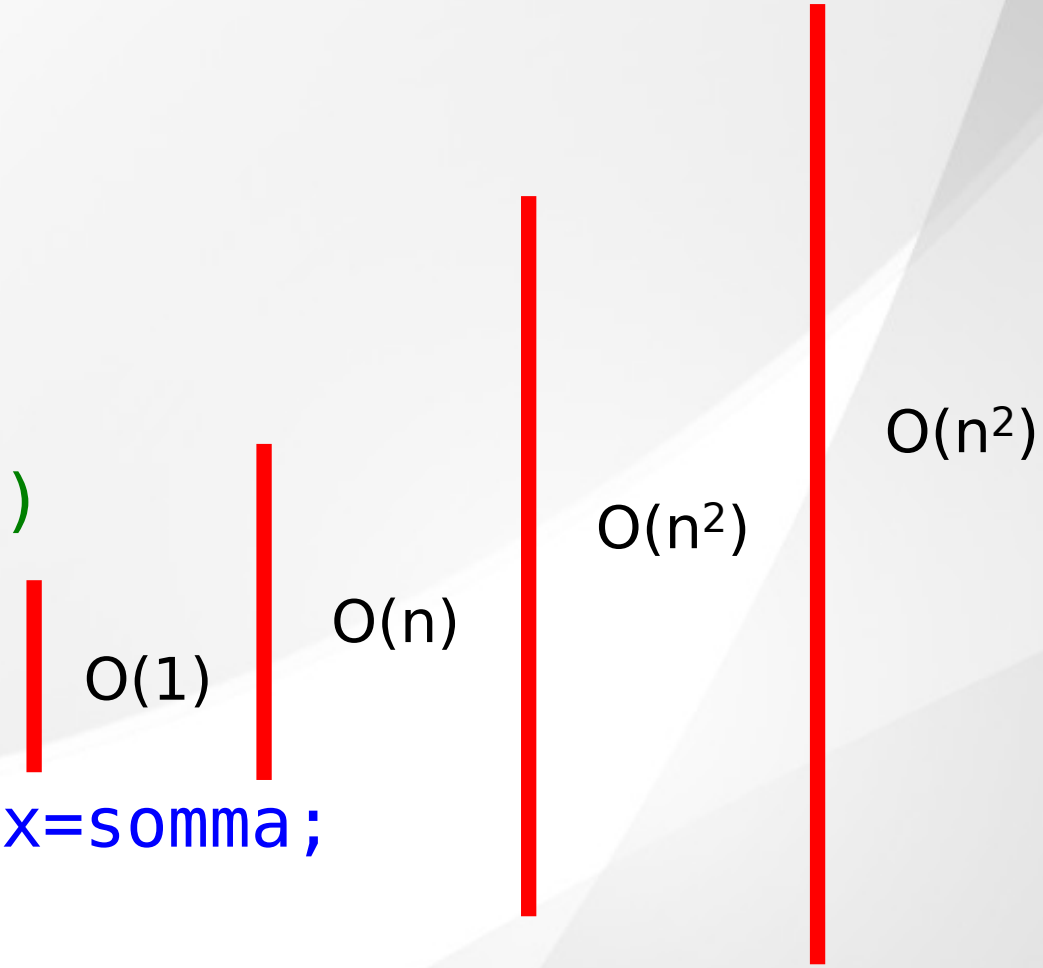
Soluzione 1

```
max=a[0];
for(i=0; i<n; i++)
{
   for(j=i; j<n; j++)
   {
      somma=0;
      for(k=i; k<=j; k++)
      {
         somma+=a[k];
      }
      if(somma > max) max=somma;
   }
}
```



Soluzione 1

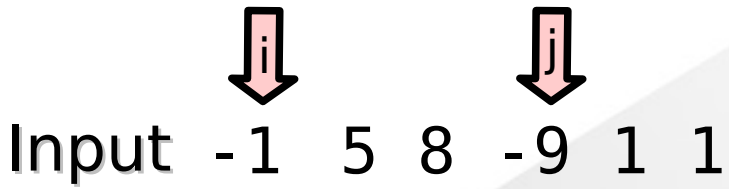
```
max=a[0];
for(i=0; i<n; i++)
{
   for(j=i; j<n; j++)
   {
      somma=0;
      for(k=i; k<=j; k++)
      {
         somma+=a[k];
      }
      if(somma > max) max=somma;
   }
}
```



Tempo: $O(n^3)$:-)

Soluzione 2

```
max=a[0];
for(i=0; i<n; i++)
{
   somma=0;
   for(j=i; j<n; j++)
   {
      somma+=a[j];
      if(somma > max) max=somma;
   }
}
```



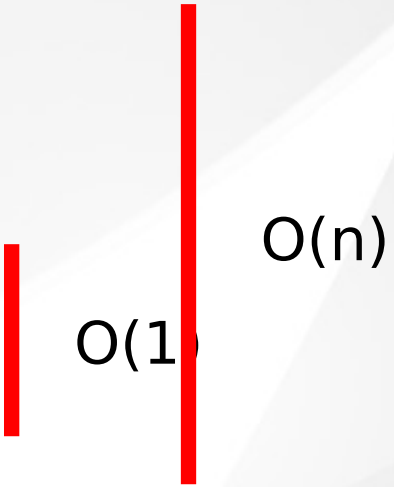
Soluzione 2

```
max=a[0];
for(i=0; i<n; i++)
{
   somma=0;
   for(j=i; j<n; j++)
   {
      somma+=a[j];
      if(somma > max) max=somma;
   }
}
```

O(1)

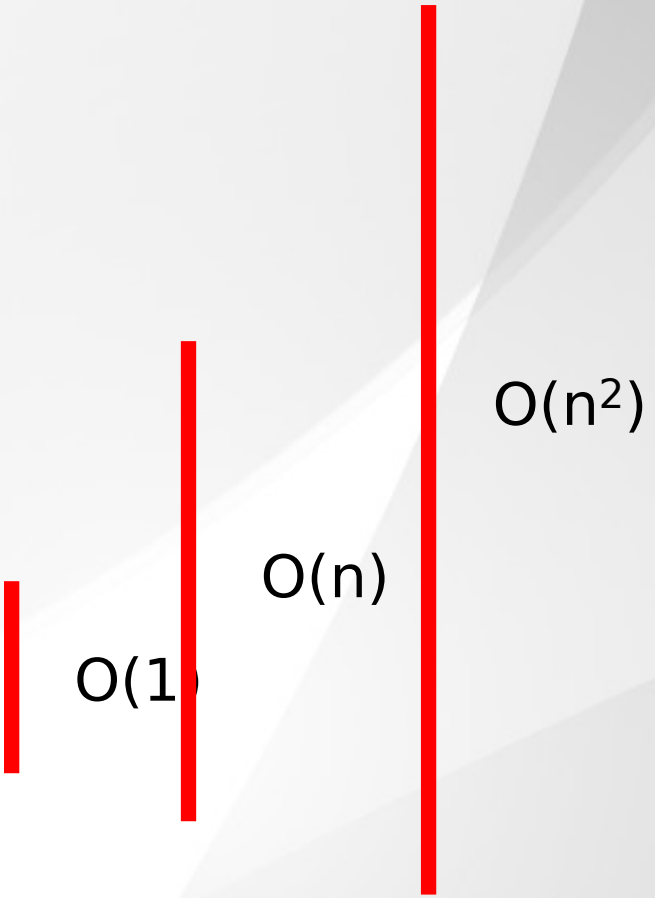
Soluzione 2

```
max=a[0];
for(i=0; i<n; i++)
{
   somma=0;
   for(j=i; j<n; j++)
   {
      somma+=a[j];
      if(somma > max) max=somma;
   }
}
```



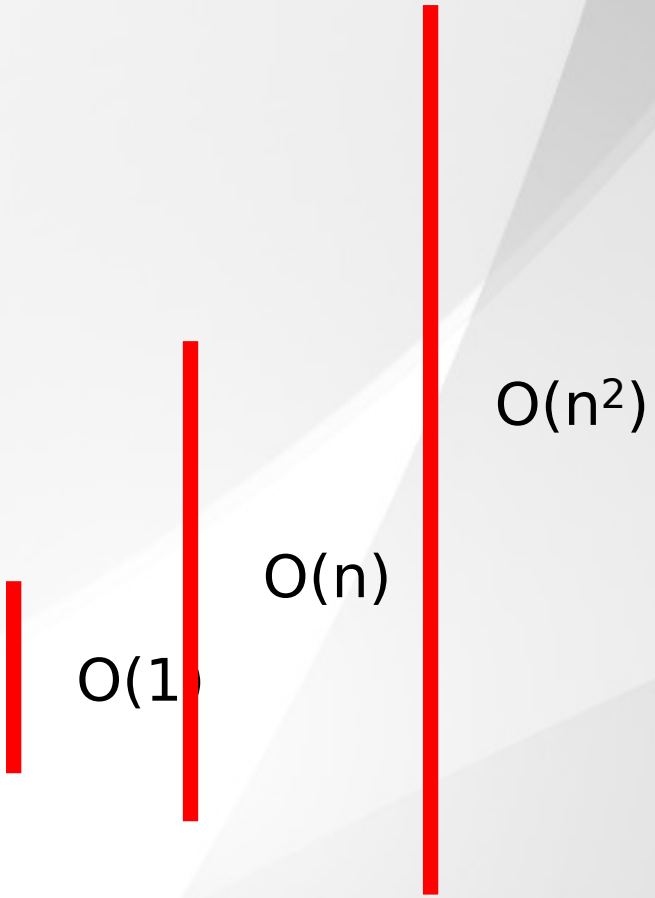
Soluzione 2

```
max=a[0];
for(i=0; i<n; i++)
{
   somma=0;
   for(j=i; j<n; j++)
   {
      somma+=a[j];
      if(somma > max) max=somma;
   }
}
```



Soluzione 2

```
max=a[0];
for(i=0; i<n; i++)
{
   somma=0;
   for(j=i; j<n; j++)
   {
      somma+=a[j];
      if(somma > max) max=somma;
   }
}
```



Tempo: $O(n^2)$:-|

Come fare meglio?

- Possiamo sfruttare due proprietà del segmento di somma massima

1) La somma dei valori in ogni prefisso del segmento ottimo è positiva, se così non fosse potremmo eliminare tale prefisso ottenendo un segmento di somma maggiore (assurdo)

2) Il valore immediatamente precedente al primo valore del segmento ottimo è negativo, se così non fosse potremmo aggiungere tale valore ottenendo un segmento di somma maggiore (assurdo)

-1 5 8 -9 1 1

Soluzione 3

```
max = A[0];
for(i=0; i<len; i++)
{
   if(somma > 0) somma+=a[i];
   else somma=a[i];

   if(somma > max) max=somma;
}
```

Tempo: O(n) :-)

Soluzione 3

```
max = A[0];
for(i=0; i<len; i++)
{
    if(somma > 0) somma+=a[i];
    else somma=a[i];

    if(somma > max) max=somma;
}
```

Algoritmo ottimo!

Perché?

Tempo: O(n) :-)

Esempio soluzione lineare

								SOMMA	MAX
1	2	-4	1	3	2	-2	1	1	1

Esempio soluzione lineare

								SOMMA	MAX
1	2	-4	1	3	2	-2	1	1	1
1	2	-4	1	3	2	-2	1	3	3

Esempio soluzione lineare

								SOMMA	MAX
1	2	-4	1	3	2	-2	1	1	1
1	2	-4	1	3	2	-2	1	3	3
1	2	-4	1	3	2	-2	1	-1	3

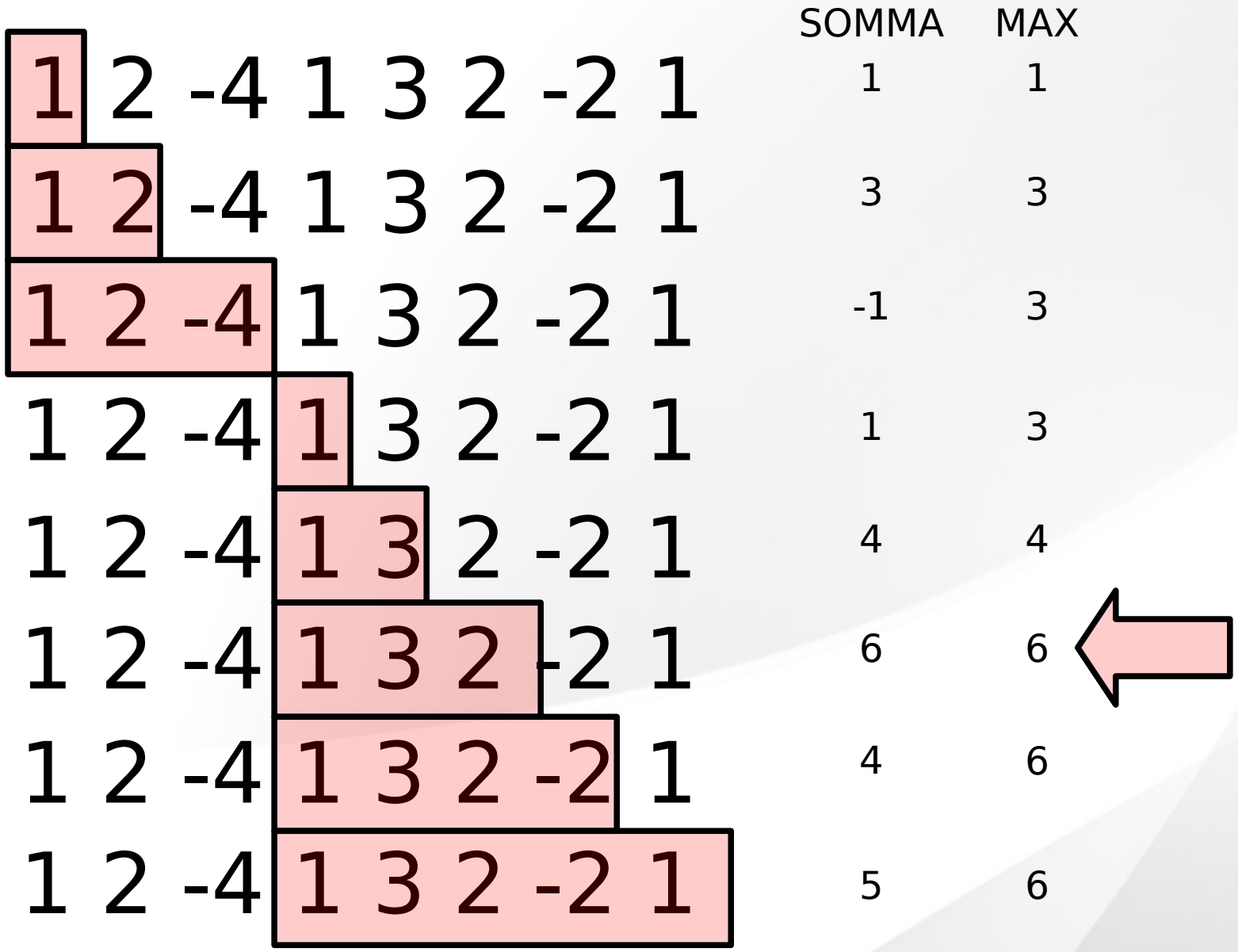
Esempio soluzione lineare

	SOMMA	MAX
1 2 -4 1 3 2 -2 1	1	1
1 2 -4 1 3 2 -2 1	3	3
1 2 -4 1 3 2 -2 1	-1	3
1 2 -4 1 3 2 -2 1	1	3

Esempio soluzione lineare

	SOMMA	MAX
1 2 -4 1 3 2 -2 1	1	1
1 2 -4 1 3 2 -2 1	3	3
1 2 -4 1 3 2 -2 1	-1	3
1 2 -4 1 3 2 -2 1	1	3
1 2 -4 1 3 2 -2 1	4	4
1 2 -4 1 3 2 -2 1	6	6
1 2 -4 1 3 2 -2 1	4	6
1 2 -4 1 3 2 -2 1	5	6

Esempio soluzione lineare



Esercizio 1

Implementare le tre soluzioni per il problema del “sottoarray di somma massima”. Per ciascuna di esse scrivere un programma che legga la sequenza di input da terminale e stampa in output il valore massimo della somma del sottoarray di somma massima.

Si assuma che la prima riga dell'input contenga la lunghezza della sequenza

INPUT 3
1
-2
4

OUTPUT 4

Provare le soluzioni sui file di input che trovate sul sito del corso, utilizzando la redirezione dell'input.

Linux: potete usare il comando time per ottenere I tempi di esecuzione del vostro programma (utile per confrontare le prestazioni).

time ./maxsum.o < input

Esercizio 2

Intersezione tra insiemi: Scrivere un programma che accetti in input dall'utente due sequenze di interi distini e stampi gli elementi che occorrono in entrambi. Si assume che l'utente specifichi la lunghezza di ogni sequenza prima dell'immissione degli elementi.