

Esercitazione su Python

Esercizio 1. Si implementi l'algoritmo di Bubble Sort sotto forma di programma procedurale in Python, utilizzando le conoscenze acquisite durante la lezione.

Il programma deve prendere la lista

`x = [3, 2, 10, 24, 1, 63, 7, 33, 45, 73]`,

e costruire la lista

`y = [1, 2, 3, 10, 24, 33, 45, 63, 73]` ordinata in maniera crescente secondo l'algoritmo summenzionato e stampare tutti gli elementi di `y` utilizzando la funzione builtin `print`.

Se possibile, fare uso dei costrutti del Python per semplificare le operazioni dell'algoritmo. Non è richiesta la scrittura di una funzione; è sufficiente una sequenza di statement.

L'algoritmo di Bubble Sort (pseudocodice) è riportato nel seguito.

Algoritmo BubbleSort

`array[elem1..elemN] = array di N elementi da ordinare`

`indice ← N - 1`

`while (indice >= 0) do`

`for i ← 0 to indice - 1 do`

`if (array[i] > array[i + 1]) then`

`swap(array[i], array[i + 1])` #non richiesto di implementarla come funzione

`indice ← indice - 1`

Esercitazione 2 su Python

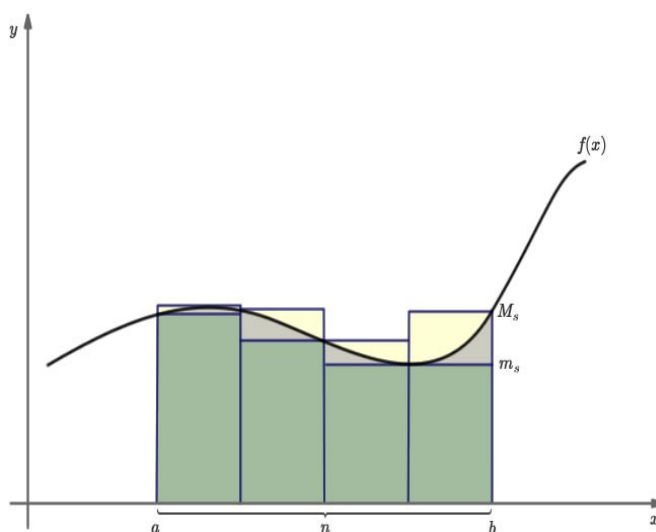
Esercizio 2. Sia data una funzione $f(x)$ continua e derivabile, e sia $[x_a, x_b]$ un intervallo su cui $f(x)$ è definita.

Si consideri il partizionamento $P = [x_a, x_{a1}, x_{a2}, \dots, x_b]$ di $[x_a, x_b]$ in un numero n di intervalli. Si definiscano, per ciascun sottointervallo $[x_j, x_{j+1}]$ del partizionamento ($j=a, \dots, b-1$), i valori $m_j = \min f(x)$ in $[x_j, x_{j+1}]$ ed $M_j = \max f(x)$ in $[x_j, x_{j+1}]$.

Le due somme di Darboux, definite di seguito:

$$s(P) = \sum m_j (x_{j+1} - x_j) \quad S(P) = \sum M_j (x_{j+1} - x_j)$$

rappresentano approssimazioni per difetto e per eccesso dell'integrale di $f(x)$ nell'intervallo $[x_a, x_b]$.



Si scriva un programma in Python che definisce: una **lista** x contenente un partizionamento dell'intervallo $[1, 3]$ in base al numero n di intervalli scelto; una lista m per gli m_j e una lista M per gli M_j ; calcoli $s(P)$, $S(P)$ per la funzione $f(x) = x^2 + 1$.

Si osservi, inoltre, che la funzione $f(x) = x^2 + 1$ è strettamente crescente; se ne deduca una opportuna semplificazione nel calcolo di m_j ed M_j .

Infine si stampino le somme calcolate $s(P)$ e $S(P)$. Si osservi il comportamento delle somme calcolate per diversi valori del numero di intervalli n .

ES.

```
# estremi dell'intervallo
```

```
xa = 1.0
```

```
xb = 3.0
```

```
# numero di elementi nel partizionamento P
```

```
n = 1000
```

```
# dimensione di ogni intervallo di P
```

```
d_x = (xb - xa) / n
```