

Pseudocodice

B. Algoritmi elementari di ordinamento

B.1. Ordinamento per selezione

Algoritmo *selectionSort* (array $A[0..n-1]$)
for $k \leftarrow 0$ **to** $n-2$ **do**
 $m \leftarrow k$
 for $j \leftarrow k+1$ **to** $n-1$ **do**
 if $A[j] < A[m]$ **then** $m \leftarrow j$
 scambia $A[m]$ con $A[k]$

B.2. Ordinamento per inserimento

Algoritmo *insertionSort* (array $A[0..n-1]$)
for $k \leftarrow 1$ **to** $n-1$ **do**
 $x \leftarrow A[k]$
 $j \leftarrow k-1$
 while $j \geq 0$ **and** $A[j] > x$ **do**
 $A[j+1] \leftarrow A[j]$
 $j \leftarrow j-1$
 $A[j+1] \leftarrow x$

B.3. Ordinamento “a bolle” (versione base)

Algoritmo *bubbleSort* (array $A[0..n-1]$)
do
 $scambiato \leftarrow false$
 for $j \leftarrow 1$ **to** $n-1$ **do**
 if $A[j] < A[j-1]$ **then**
 scambia $A[j-1]$ con $A[j]$
 $scambiato \leftarrow true$
while $scambiato$

B.4. Ordinamento “a bolle” (versione migliorata)

All'iterazione i esamina solo gli elementi fino alla posizione $n - i$ anziché fino all'ultima posizione. Infatti, gli elementi nelle posizioni di indice $> n - i$ sono già stati collocati nella posizione finale durante le iterazioni precedenti. La condizione $i < n$ per il ciclo esterno evita un'ulteriore scansione dell'array nel caso nell'iterazione $n - 1$ vi siano stati scambi. Infatti dopo l'iterazione $n - 1$ l'array sarà in ogni caso ordinato.

Algoritmo *bubbleSort* (array $A[0..n - 1]$)

```
 $i \leftarrow 1$ 
do
  |  $scambiato \leftarrow false$ 
  | for  $j \leftarrow 1$  to  $n - i$  do
  |   | if  $A[j] < A[j - 1]$  then
  |   |   | scambia  $A[j - 1]$  con  $A[j]$ 
  |   |   |  $scambiato \leftarrow true$ 
  |  $i \leftarrow i + 1$ 
while  $scambiato$  and  $i < n$ 
```
