

## Selection sort

Il selection sort è un algoritmo di ordinamento molto poco efficiente, benché naturale:

→ Estrae il minimo, lo muove a l'inizio e continua sul resto dell'array

```
for (int i=0; i<n-1; i++) {  
    int min = i;
```

```
    for (int j=i+1; j<n; j++) {  
        if (A[j] < A[min]) {  
            min = j;  
        }  
    }
```

$O(n^2)$ ,  $\Omega(n^2)$ ,  
 $\Theta(n^2)$

```
    int c = A[i];  
    A[i] = A[min];  
    A[min] = c;  
}
```

sempre  $n^2$ , anche  
se  $A$  è già  
ordinato...

## Insertion sort

L'idea dell'insertion sort è ancora naturale, ma più efficiente di quella del selection sort nel caso pessimo.

→ A ogni  $i$ -esimo passo,  $A[0-i]$  è ordinato, si prende  $A[i+1]$  e lo si ordina da l'alto in  $A[0-i+1]$ .

```
for (int i=1; i < n; i++) {  
    for (int j=i-1; j >= 0 && A[j+1] <  
        A[j]; j--) {  
        int c = A[j+1];  
        A[j+1] = A[j];  
        A[j] = c;  
    }  
}
```

→ se  $A$  è già ordinato, richiede solo una scansione, ossia  $\Omega(n)$ ; altrimenti  $O(n^2)$ .