Selection Sort
Selection Sort

• In selection sort, the idea of the algorithm is to find the smallest unsorted element and add it to the end of the sorted list.

In pseudocode:
• Repeat until no unsorted elements remain:
  • Search the unsorted part of the data to find the smallest value
  • Swap the smallest found value with the first element of the unsorted part
Selection Sort

In pseudocode:
Repeat until no unsorted elements remain:
   Search the unsorted part of the data to find the smallest value
   Swap the smallest found value with the first element of the unsorted part
Selection Sort

5 2 1 3 6 4

In pseudocode:
Repeat until no unsorted elements remain:
    Search the unsorted part of the data to find the smallest value
    Swap the smallest found value with the first element of the unsorted part
Selection Sort

In pseudocode:
Repeat until no unsorted elements remain:
   Search the unsorted part of the data to find the smallest value
   Swap the smallest found value with the first element of the unsorted part
Selection Sort

In pseudocode:
Repeat until no unsorted elements remain:
   Search the unsorted part of the data to find the smallest value
   Swap the smallest found value with the first element of the unsorted part
Selection Sort

In pseudocode:
Repeat until no unsorted elements remain:
Search the unsorted part of the data to find the smallest value
Swap the smallest found value with the first element of the unsorted part
Selection Sort

In pseudocode:
Repeat until no unsorted elements remain:
   Search the unsorted part of the data to find the smallest value
   Swap the smallest found value with the first element of the unsorted part
Selection Sort

In pseudocode:
Repeat until no unsorted elements remain:
Search the unsorted part of the data to find the smallest value
Swap the smallest found value with the first element of the unsorted part
Selection Sort

In pseudocode:
Repeat until no unsorted elements remain:
   Search the unsorted part of the data to find the smallest value
   Swap the smallest found value with the first element of the unsorted part
Selection Sort

In pseudocode:

Repeat until no unsorted elements remain:
  Search the unsorted part of the data to find the smallest value
  Swap the smallest found value with the first element of the unsorted part
Selection Sort

In pseudocode:
Repeat until no unsorted elements remain:
    Search the unsorted part of the data to find the smallest value
    Swap the smallest found value with the first element of the unsorted part
Selection Sort

In pseudocode:
Repeat until no unsorted elements remain:
    Search the unsorted part of the data to find the smallest value
    Swap the smallest found value with the first element of the unsorted part
Selection Sort

In pseudocode:
Repeat until no unsorted elements remain:
  Search the unsorted part of the data to find the smallest value
  Swap the smallest found value with the first element of the unsorted part
Selection Sort

In pseudocode:
Repeat until no unsorted elements remain:
  Search the unsorted part of the data to find the smallest value
  Swap the smallest found value with the first element of the unsorted part
In pseudocode:
Repeat until no unsorted elements remain:
  Search the unsorted part of the data to find the smallest value
  Swap the smallest found value with the first element of the unsorted part

Selection Sort
Selection Sort

In pseudocode:
Repeat until no unsorted elements remain:
   Search the unsorted part of the data to find the smallest value
   Swap the smallest found value with the first element of the unsorted part
Selection Sort

In pseudocode:
Repeat until no unsorted elements remain:
  Search the unsorted part of the data to find the smallest value
  Swap the smallest found value with the first element of the unsorted part
## Selection Sort

| 1 | 2 | 3 | 4 | 5 | 6 |

**In pseudocode:**

Repeat until no unsorted elements remain:

1. Search the unsorted part of the data to find the smallest value
2. Swap the smallest found value with the first element of the unsorted part
Selection Sort

• **Worst-case scenario**: We have to iterate over each of the $n$ elements of the array (to find the smallest unsorted element) and we must repeat this process $n$ times, since only one element gets sorted on each pass.

• **Best-case scenario**: Exactly the same! There’s no way to guarantee the array is sorted until we go through this process for all the elements.
Selection Sort

$O(n^2)$

$\Omega(n^2)$