

# Chapter 38: Selection Sort

## Section 38.1: Elixir Implementation

```
defmodule Selection do

  def sort(list) when is_list(list) do
    do_selection(list, [])
  end

  def do_selection([head|[]], acc) do
    acc ++ [head]
  end

  def do_selection(list, acc) do
    min = min(list)
    do_selection(:lists.delete(min, list), acc ++ [min])
  end

  defp min([first|[second|[]]]) do
    smaller(first, second)
  end

  defp min([first|[second|tail]]) do
    min([smaller(first, second)|tail])
  end

  defp smaller(e1, e2) do
    if e1 <= e2 do
      e1
    else
      e2
    end
  end
end

Selection.sort([100, 4, 10, 6, 9, 3])
|> IO.inspect
```

## Section 38.2: Selection Sort Basic Information

[Selection sort](#) is a sorting algorithm, specifically an in-place comparison sort. It has  $O(n^2)$  time complexity, making it inefficient on large lists, and generally performs worse than the similar insertion sort. Selection sort is noted for its simplicity, and it has performance advantages over more complicated algorithms in certain situations, particularly where auxiliary memory is limited.

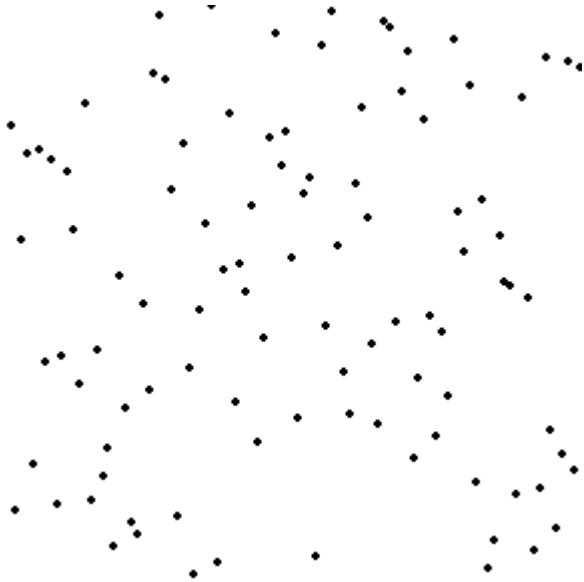
The algorithm divides the input list into two parts: the sublist of items already sorted, which is built up from left to right at the front (left) of the list, and the sublist of items remaining to be sorted that occupy the rest of the list. Initially, the sorted sublist is empty and the unsorted sublist is the entire input list. The algorithm proceeds by finding the smallest (or largest, depending on sorting order) element in the unsorted sublist, exchanging (swapping) it with the leftmost unsorted element (putting it in sorted order), and moving the sublist boundaries one element to the right.

### Pseudo code for Selection sort:

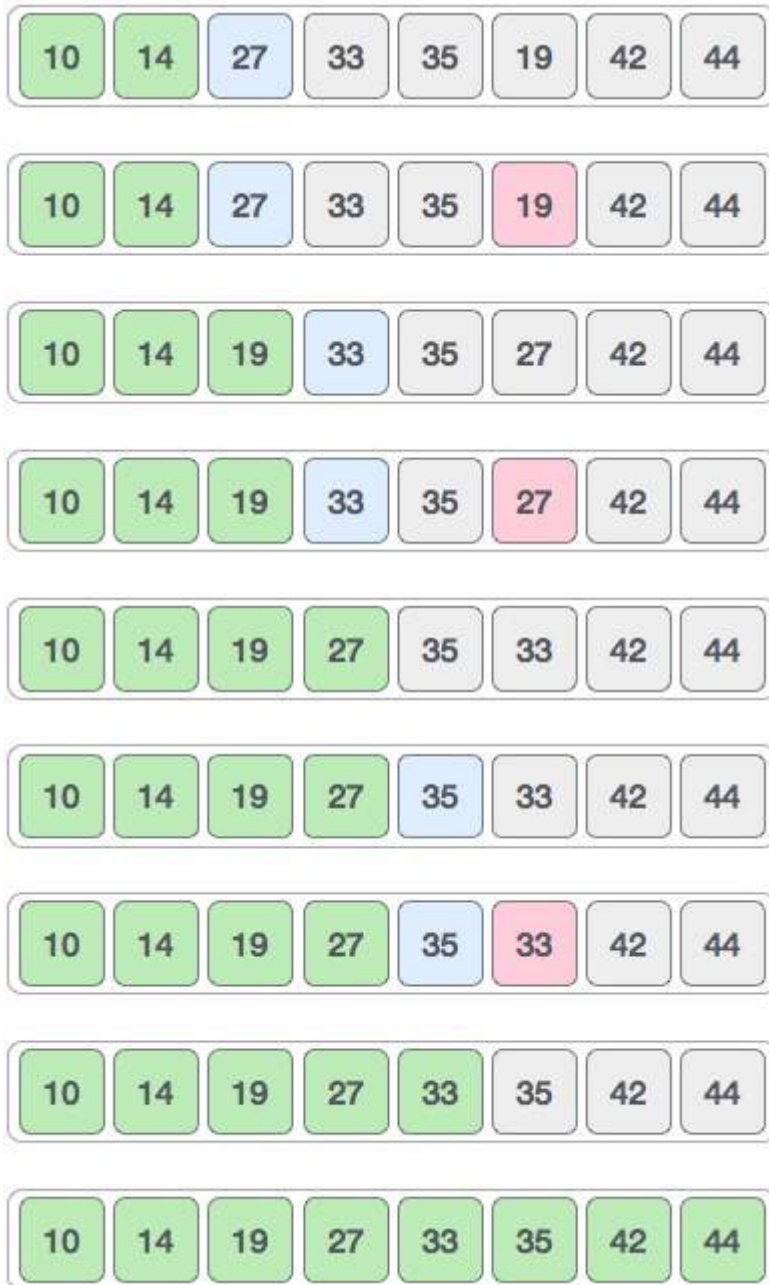
```
function select(list[1..n], k)
  for i from 1 to k
```

```
minIndex = i
minValue = list[i]
for j from i+1 to n
    if list[j] < minValue
        minIndex = j
        minValue = list[j]
swap list[i] and list[minIndex]
return list[k]
```

### Visualization of selection sort:



### Example of Selection sort:



**Auxiliary Space:**  $O(n)$

**Time Complexity:**  $O(n^2)$

## Section 38.3: Implementation of Selection sort in C#

I used C# language to implement Selection sort algorithm.

```
public class SelectionSort
{
    private static void SortSelection(int[] input, int n)
    {
        for (int i = 0; i < n - 1; i++)
        {
            var minId = i;
            int j;
            for (j = i + 1; j < n; j++)
            {
                if (input[j] < input[minId]) minId = j;
            }
            var temp = input[minId];
        }
    }
}
```

```
        input[minId] = input[i];  
        input[i] = temp;  
    }  
}  
  
public static int[] Main(int[] input)  
{  
    SortSelection(input, input.Length);  
    return input;  
}  
}
```