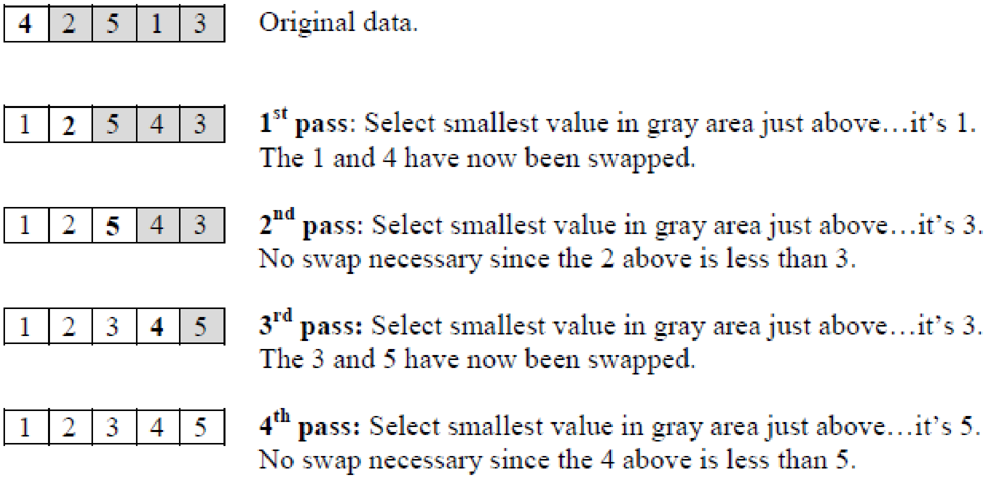
Selection Sort $O(n^2)$ for all cases

The Selection Sort uses an **incremental** approach. During the first pass the smallest value is **selected** from the entire array and swapped with the first element. On the second pass the smallest value is **selected** from the array beginning with the 2nd element and swapped with the second element, etc….the above description is for an ascending sort. The following shows the sequence of steps in a Selection Sort:



4	2	5	1	3	Original data.
1	**2**	5	4	3	1st **pass**: Select smallest value in gray area just above…it's 1. The 1 and 4 have now been swapped.
1	2	**5**	4	3	2nd **pass**: Select smallest value in gray area just above…it's 3. No swap necessary since the 2 above is less than 3.
1	2	3	**4**	5	3rd **pass:** Select smallest value in gray area just above…it's 3. The 3 and 5 have now been swapped.
1	2	3	4	5	4th **pass:** Select smallest value in gray area just above…it's 5. No swap necessary since the 4 above is less than 5.

A Selection Sort method:

```
public static void sort(int a[ ])
{
        int min, minIndex;
        for(int i = 0;i < a.length; ++i)
        {
                min = a[i];
                minIndex = i;
                for (int j = i + 1; j < a.length; ++j) // Find minimum
                {
                        if (a[j] < min) //salient feature
                        {
                                min = a[j];
                                minIndex = j;
                        }
                }
                a[minIndex] = a[i]; // swap
                a[i] = min;
        }
}
```

Disadvantage:

A disadvantage of the selection sort is that it will not allow an early exit from the entire process if the list becomes ordered in an early pass.