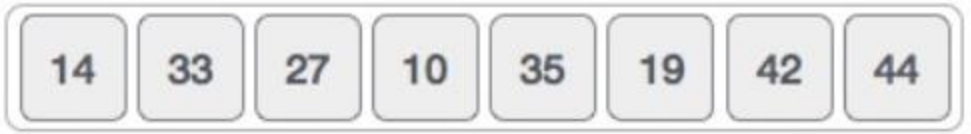
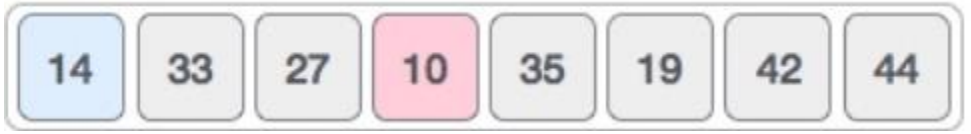
Selection Sort

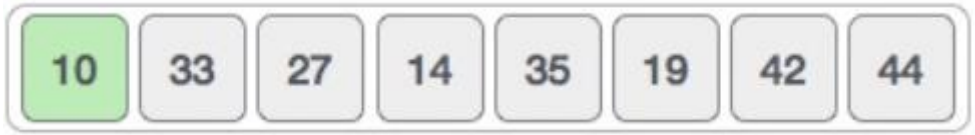
- Selection sort is a simple sorting algorithm in which the list is divided into two parts, the sorted part at the left end and the unsorted part at the right end.
- Initially, the sorted part is empty and the unsorted part is the entire list.
- The smallest element is selected from the unsorted array and swapped with the leftmost element, and that element becomes a part of the sorted array.
- This process continues moving unsorted array boundary by one element to the right.



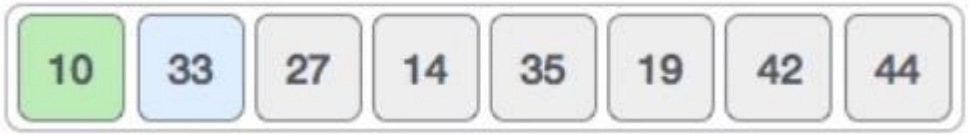
- For the first position in the sorted list, the whole list is scanned sequentially.
- Search the whole list and find the lowest value.



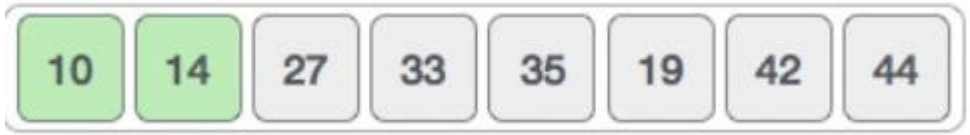
- After one iteration 10, which happens to be the minimum value in the list, appears in the first position of the sorted list.



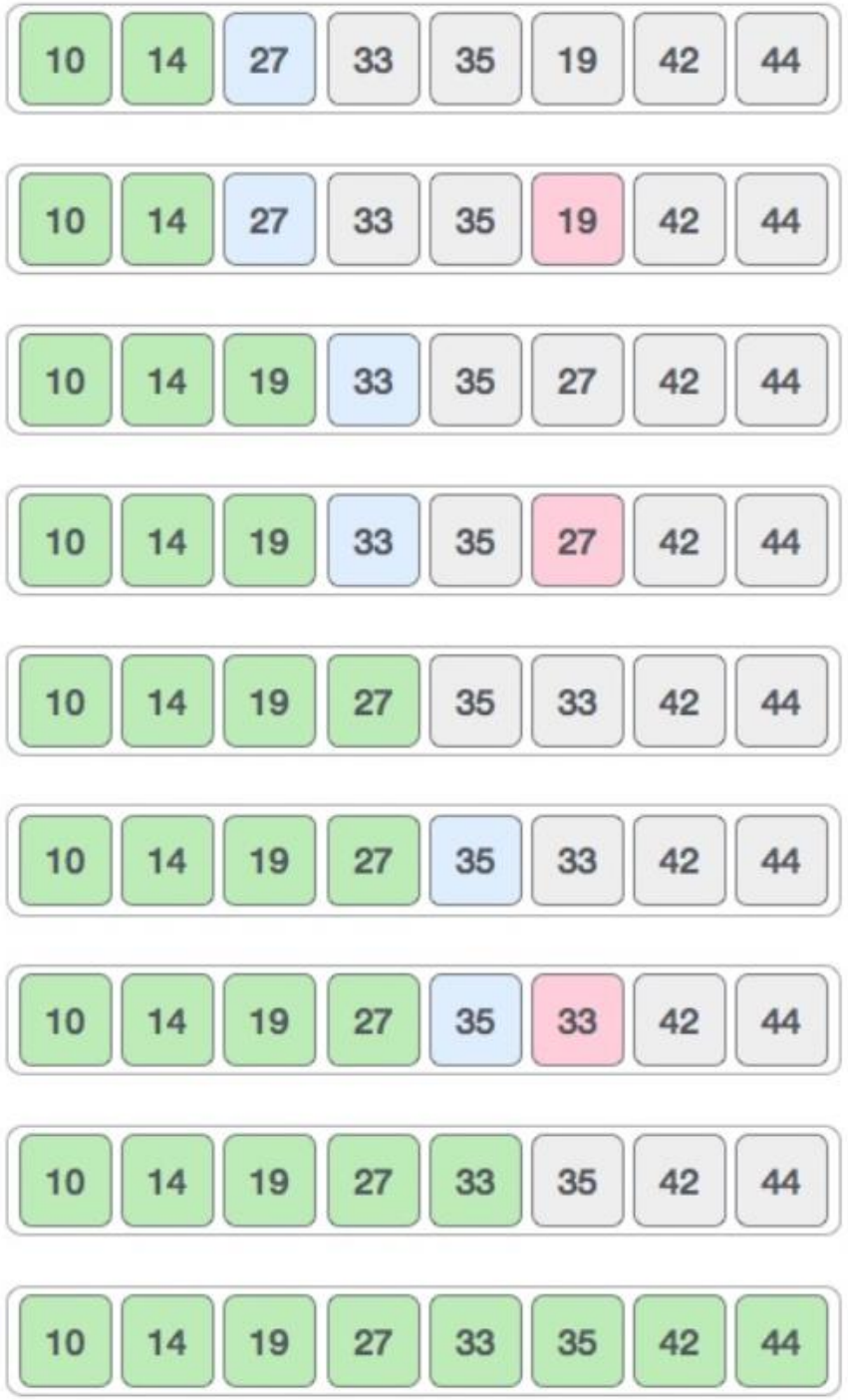
- For the second position, where 33 is residing, start scanning the rest of the list in a linear manner.



- 14 is the second lowest value in the list and it should appear at the second place. Swap these values.



- After two iterations, two least values are positioned at the beginning in a sorted manner.




```
Step 1 - Set MIN to location 0
Step 2 - Search the minimum element in the list
Step 3 - Swap with value at location MIN
Step 4 - Increment MIN to point to next element
Step 5 - Repeat until list is sorted
```

A=[] : array to be sorted
n= size of array

```
for  i ← 0 to n-2
{
 min=i
     for j ← i+1 to n-1
      {
         if (A[j]<A[min])
          min= j
      }
 temp=A[i]
A[i]=A[min]
A[min]=temp

}
```