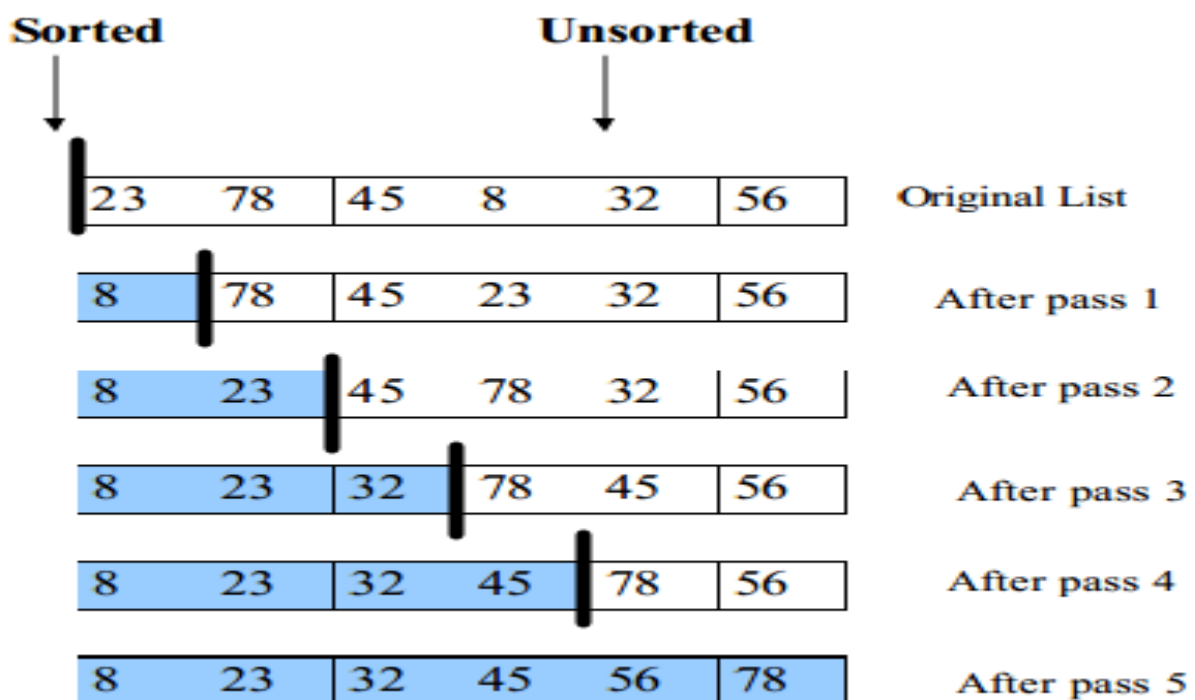


Selection Sort

- ० सूची, दो sublists, हल और unsorted में विभाजित है एक काल्पनिक दीवार से विभाजित कर रहे हैं.
- ० हम unsorted sublist से छोटी तत्व खोजने और unsorted डेटा की शुरुआत में तत्व के साथ यह स्वैप.
- ० प्रत्येक चयन और गमागमन, काल्पनिक दीवार के बाद दो sublists, आगे एक तत्व ले जाने के बीच सॉर्ट किए गए तत्वों की संख्या बढ़ रही है और कम unsorted लोगों की संख्या.
- ० हम unsorted से एक तत्व के लिए कदम हर बार सॉर्ट किया गया sublist को sublist, हम हम कहते हैं कि एक तरह से पास पूरा किया.
- ० N तत्वों की सूची N-1 के लिए पूरी तरह से गुजरता की आवश्यकता डेटा को पुनर्व्यवस्थित



CODES{C}

```
#include <stdio.h>

int main()
{
    int array[100], n, c, d, position, swap;

    printf("Enter number of elements\n");
    scanf("%d", &n);

    printf("Enter %d integers\n", n);

    for ( c = 0 ; c < n ; c++ )
        scanf("%d", &array[c]);

    for ( c = 0 ; c < ( n - 1 ) ; c++ )
    {
        position = c;

        for ( d = c + 1 ; d < n ; d++ )
        {
            if ( array[position] > array[d] )
                position = d;
        }
        if ( position != c )
        {
            swap = array[c];
            array[c] = array[position];
            array[position] = swap;
        }
    }

    printf("Sorted list in ascending order:\n");

    for ( c = 0 ; c < n ; c++ )
        printf("%d\n", array[c]);

    return 0;
}
```

CODES{JAVA}

```
public class MySelectionSort {  
  
    public static int[] doSelectionSort(int[] arr){  
  
        for (int i = 0; i < arr.length - 1; i++)  
        {  
            int index = i;  
            for (int j = i + 1; j < arr.length; j++)  
                if (arr[j] < arr[index])  
                    index = j;  
  
            int smallerNumber = arr[index];  
            arr[index] = arr[i];  
            arr[i] = smallerNumber;  
        }  
        return arr;  
    }  
  
    public static void main(String a[]){  
  
        int[] arr1 = { 10,34,2,56,7,67,88,42};  
        int[] arr2 = doSelectionSort(arr1);  
        for(int i:arr2){  
            System.out.print(i);  
            System.out.print(", ");  
        }  
    }  
}
```