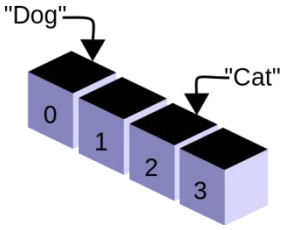


Data Structure

&

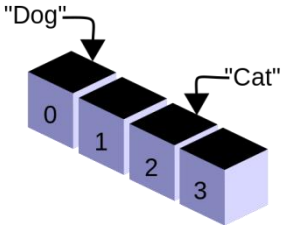
Algorithm

Class X

Lab 10

UNITED TRUST

bdeducation
Bangladesh Digital Education Research Limited
সময়ের পাঠশালা

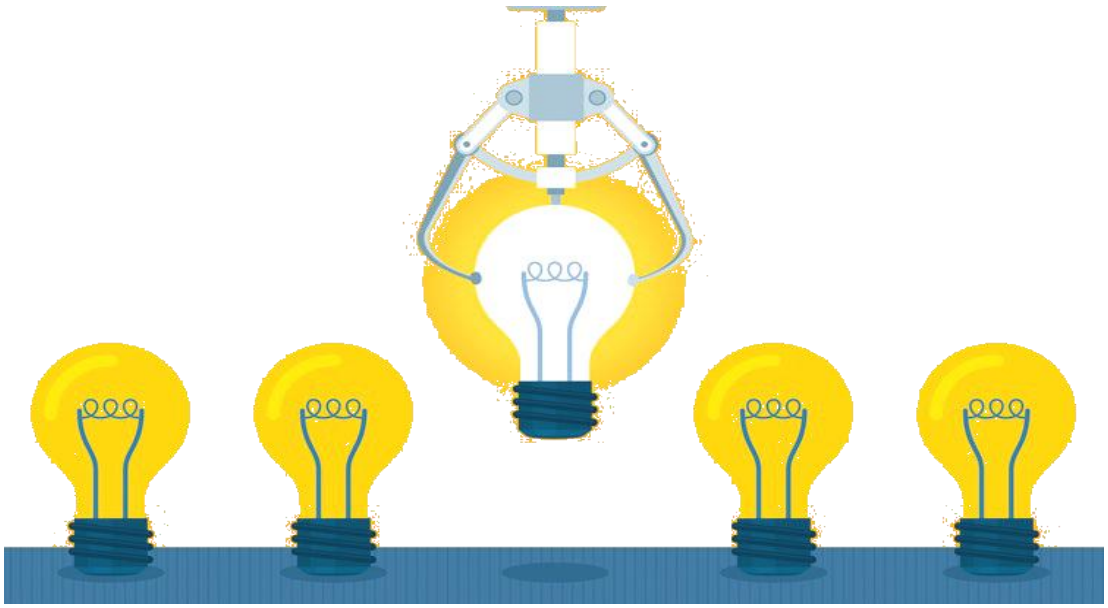


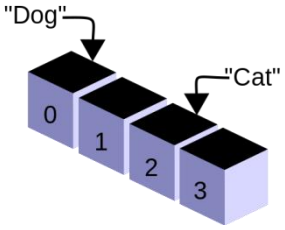
Lab Objectives:

- Selection Sort

Selection Sort

Selection sort is a simple sorting algorithm. This sorting algorithm is an in-place comparison-based 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.





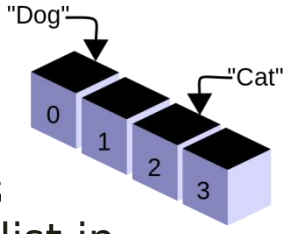
The smallest element is selected from the unsorted array and swapped with the leftmost element, and that element becomes a part of the sorted array. This process continues moving unsorted array boundary by one element to the right.

How Selection Sort Works?

1) Consider the following depicted array as an example.

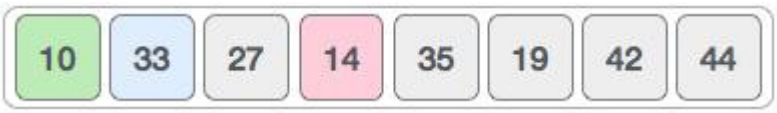
2) For the first position in the sorted list, the whole list is scanned sequentially. The first position where 14 is stored presently, we search the whole list and find that 10 is the lowest value.

3) So we replace 14 with 10. After one iteration 10, which happens to be the minimum value in the list, appears in the first position of the sorted list.



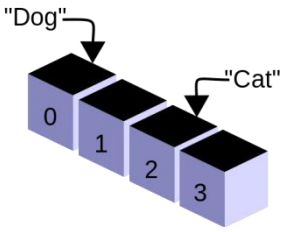
4) For the second position, where 33 is residing, we start scanning the rest of the list in a linear manner.

5) We find that 14 is the second lowest value in the list and it should appear at the second place. We swap these values.



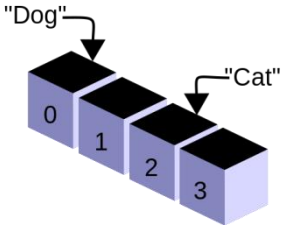
6) After two iterations, two least values are positioned at the beginning in a sorted manner.

7) The same process is applied to the rest of the items in the array.



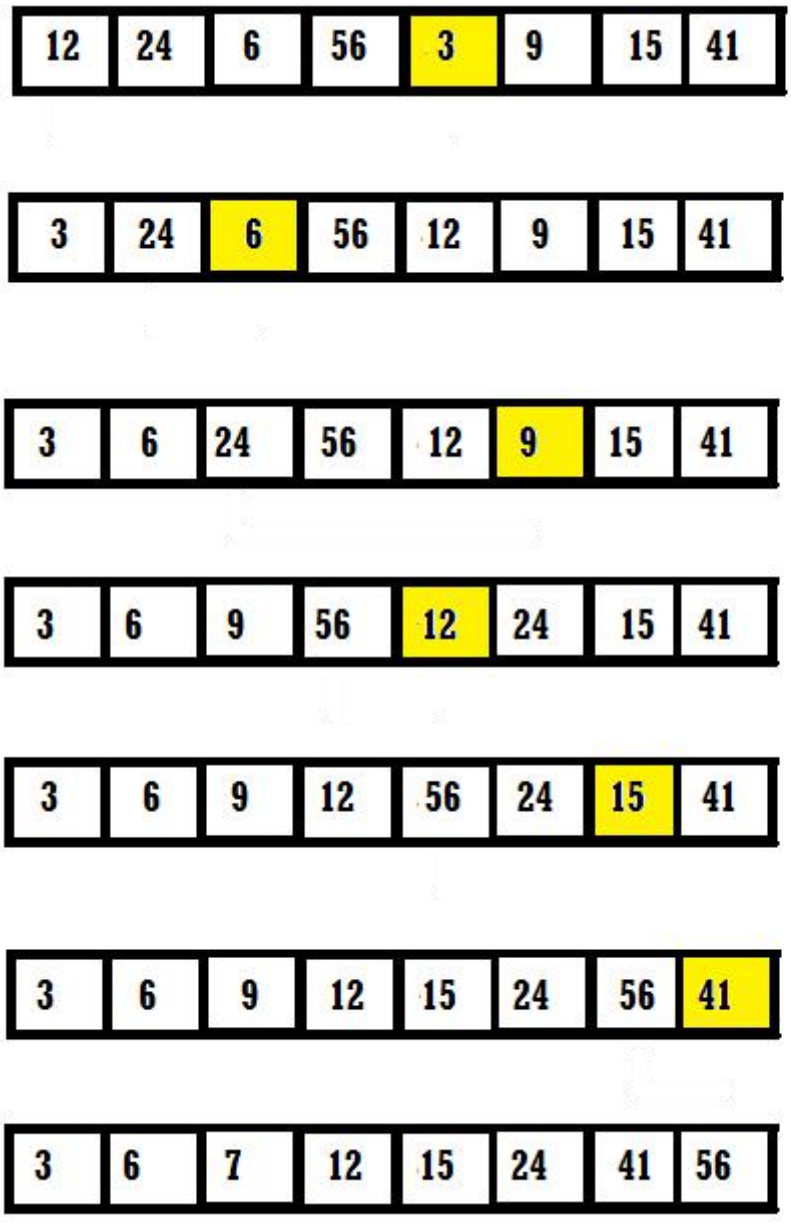
8) Following is a pictorial depiction of the entire sorting process −

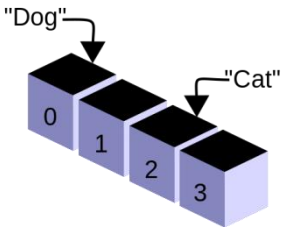
Now, let us learn some programming aspects of selection sort.



Example 1: Decorate step by step into Selection Sort

12	24	6	56	3	9	15	41





Example 2: Decorate step by step into Selection Sort

8	5	7	1	9	3

Selection Sort.

comparisons

8 5 7 1 9 3 — (n - 1) first smallest

1 5 7 8 9 3 — (n - 2) second smallest

1 3 7 8 9 5 — (n - 3) third smallest

1 3 5 8 9 7 — 2

1 3 5 7 9 8 — 1

1 3 5 7 8 9 — 0

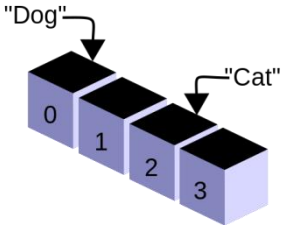
Sorted List.

Current.

Exchange.

Total comparisons = $n(n-1)/2$

$\sim O(n^2)$



Algorithm

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

Pseudocode

```
procedure selection sort
   list  : array of items
   n     : size of list

   for i = 1 to n - 1
   /* set current element as minimum*/
      min = i

      /* check the element to be minimum */

      for j = i+1 to n
         if list[j] < list[min] then
            min = j;
         end if
      end for

      /* swap the minimum element with the current element*/
      if indexMin != i  then
         swap list[min] and list[i]
      end if
   end for

end procedure
```