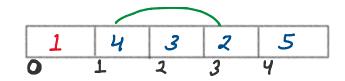
23/09/2023

SELECTION SORT

SORTINGS 02: SELECTION SORT - What if, I select the minimum element and put it at right position - For ith iteration, pick smallest element from i to (n-1) index and swap it with ith element 1 2 ass After sorting array in ascending Min Indux = 1 Stanting mu Finst Eliment Ko small ust maan lo SWAP 1st Iteration ▶Step 01: find smallest element's index from [0,N-1] U COM j=0 j=4 min Indux = j Step 02: swap the ith and minIndex position's value Swap (au [17, au [Min Indux])
5 1

2nd Iteration

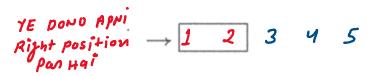


3 com

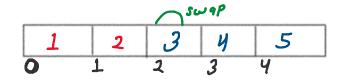
▷Step 01: find smallest element's index from [0,N-1]

$$j=1$$
 $j=3$ min Indux = $j=3$

Step 02: swap the ith and minIndex position's value



3rd Iteration



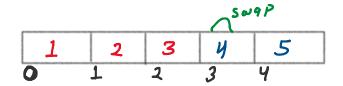
2 GM

▷ Step 01: find smallest element's index from [0,N-1]

$$j=2$$
 $j=2$ min Indux = $j=2$

1 2 3 U 5

4rt Iteration

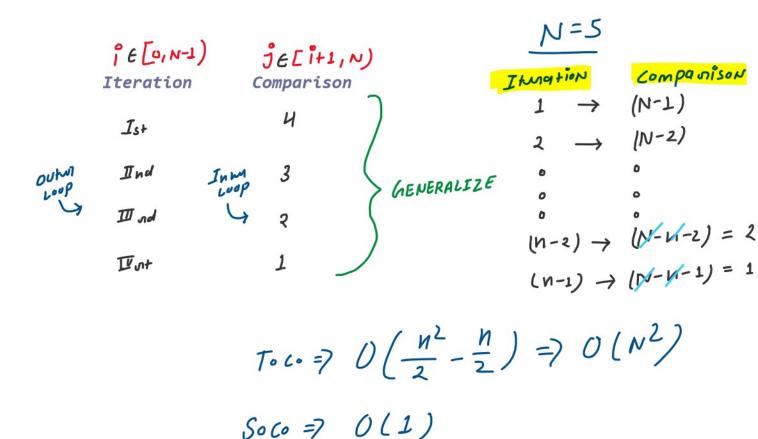


▷Step 01: find smallest element's index from [0,N-1]

$$j=3$$
 $j=3$ min Indux = $j=3$

Sorted Array





```
// SORTINGS 02: SELECTION SORT
#include<vector>
using namespace std;

// SELECTION SORT Function
void selectionSort(vector<int> &arr){
  int N = arr.size();

  for(int i=0; i < N-1; i++){

    // Lets assume ki 1st index par jo value hai wo hi smallest hai
    int minIndex = i;

    // >Step 01: find smallest element's index from [0,N-1]
    for(int j = i+1; j < N; j++){

        if(arr[j] < arr[minIndex]){
            minIndex = j;
        }
    }
    // >Step 02: swap the ith and minIndex position's value
    swap(arr[i], arr[minIndex]);
```

```
minindex = ];
}
}
// ▶Step 02: swap the ith and minindex position's value
swap(arr[i], arr[minIndex]);
}

int main(){
    vector<int> arr{5,4,3,1,2};

    selectionSort(arr);
    for(auto value: arr){
        cout<<value<<" ";
    }
    return 0;
}

/*
INPUT: {5,4,3,1,2}
OUTPUT: {1,2,3,4,5}
TIME COMPLEXITY: 0(1)
*/

*/
```