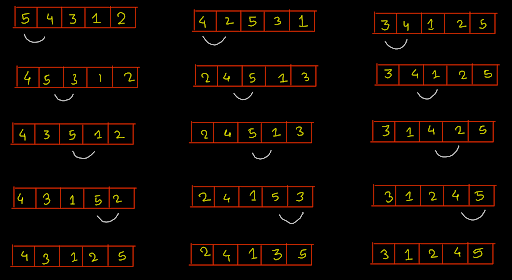
**1. PROBLEM DISCUSSION**

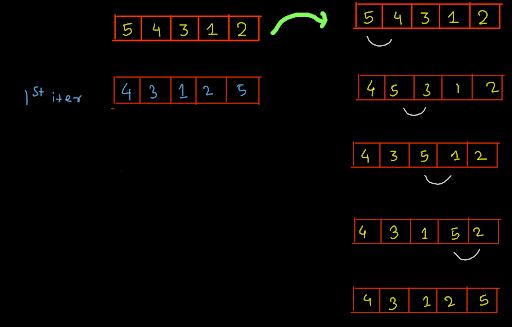
In this problem we will be given an array of integers. We have to sort the array in increasing order using Bubble sort. Make sure to use the swap and compare function which are provided, to match the output.

**2. APPROACH**

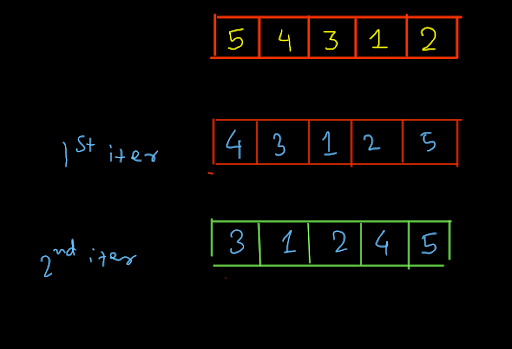
So we will be comparing each element with its next adjacent element and swap the larger element to the right. This will ensure that after one pass largest element will be present in the rightmost index. Lets verify this



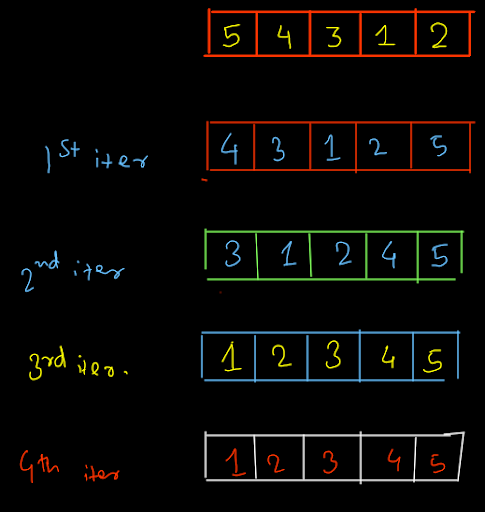
We have taken three cases here When the largest element was present at the first index When the largest element was present in somewhere middle When the largest element was present in the last index In all the cases after swapping the largest element got shifted to the last index. So for a given array of length N we will apply the above idea N-1 times,each time the largest element will get shifted to the right .



So we compared the adjacent elements and swapped the larger element to the right. After the 1st iteration the largest element got shifted to the right. Now again we will iterate on the array and again compare the element and swap the elements.



So we can see after 2 iterations the second largest element got shifted to right in its correct position. So we will repeat the above process for N-1 times where N is the length of array.



ConsoleCpp

#include<iostream>

using namespace std;

bool isSmaller(int arr[], int i, int j ) {

cout << "Comparing " << arr[i] << " and " << arr[j] << endl;

if (arr[i] < arr[j]) {

return true;

} else {

return false;

}

}

void swap(int arr[], int i, int j) {

cout << "Swapping " << arr[i] << " and " << arr[j] << endl;

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

void bubbleSort(int arr[], int n) {

for (int i = 0; i < n; i++) {

for (int c = 1; c < n - i; c++) {

int p = c - 1;

if (isSmaller(arr, c, p)) {

swap(arr, c, p);

}

}

}

}

void print(int arr[], int n) {

for (int i = 0; i < n; i++) {

cout << arr[i] << endl;

}

}

int main() {

int n;

cin >> n;

int arr[n];

for (int i = 0; i < n; i++) {

cin >> arr[i];

}

bubbleSort(arr, n);

print(arr, n);

}

**3. ANALYSIS**

Time complexity O( N^2 ). Space complexity s=O(1)