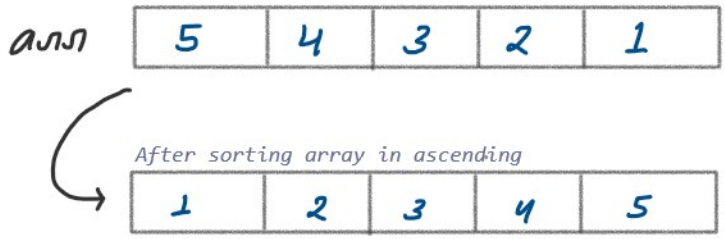
BUBBLE SORT

24 September 2023 09:22

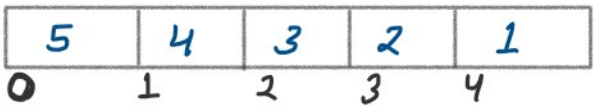
23/09/2023

BUBBLE SORT

✅**SORTINGS 01:** BUBBLE SORT
- swap the adjacent if needed till we get all the array sorted

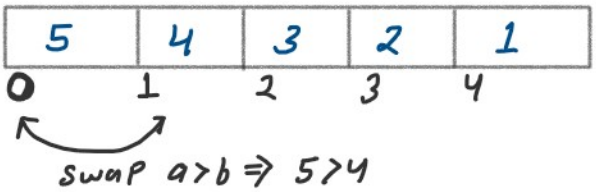


DRY RUN



1st Iteration

1st largest element at it's positions



a
b
} Adjacent value

a > b → swap

a < b → NO swap

4 5 3 2 1

swap a>b ⇒ 5>3

4 3 5 2 1

swap a>b ⇒ 5>2

4 3 2 5 1

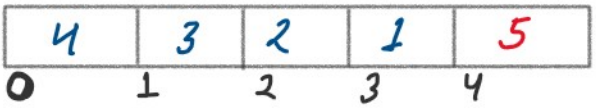
swap a>b ⇒ 5>1

4 3 2 1 5

1st largest element = 5

2nd Iteration

2nd largest element at it's positions



swap a>b => 4>3

3 4 2 1

sawap a>b => 4>2

3 2 4 1

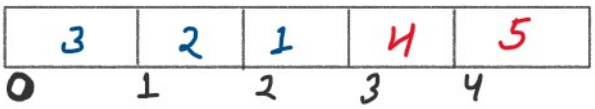
swap a>b => 4>1

3 2 1 4

2nd largest element = 4

3rd Iteration

3rd largest element at it's positions



swap a>b => 3>2

2 3 1

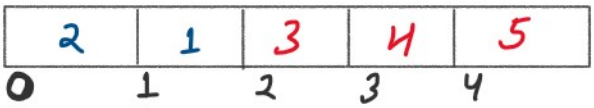
swap a>b => 3>1

2 1 3

3rd largest element = 3

4rt Iteration

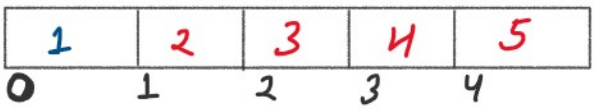
4rt largest element at it's positions



swap a>b => 2>1

1 2

4rt largest element = 2

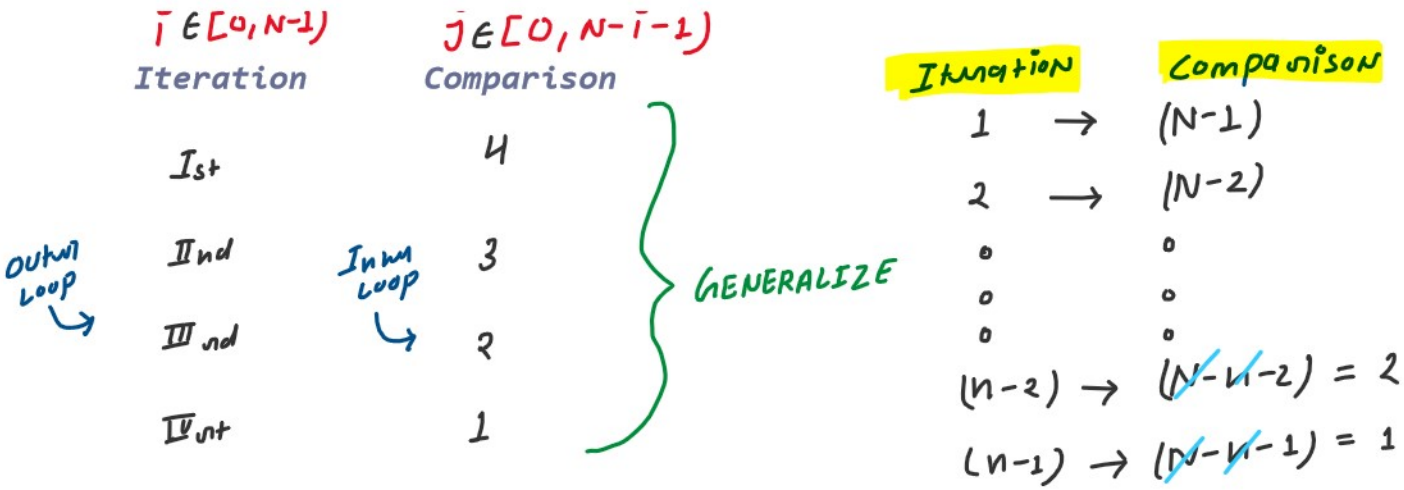


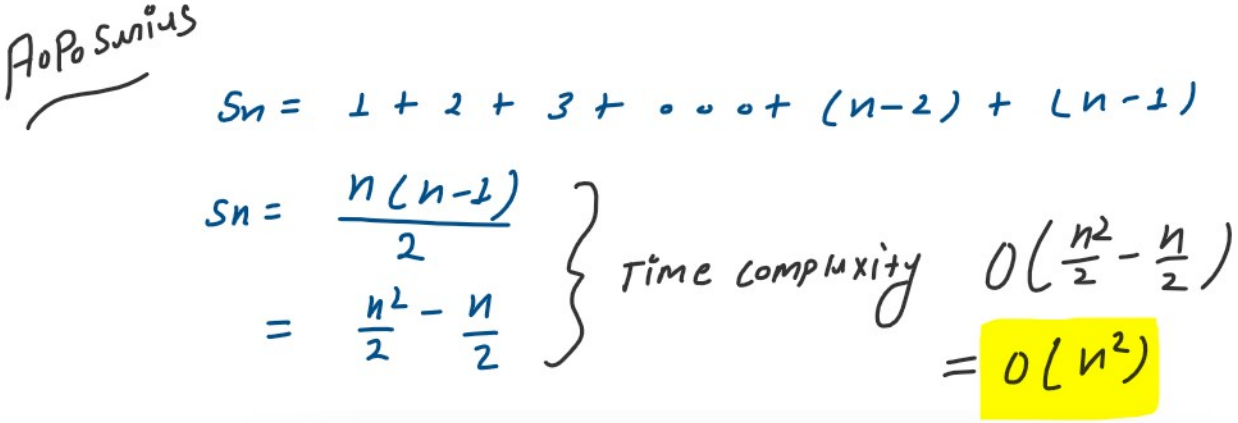
N = size = 5

$i \in [0, N-1)$ Iteration

$j \in [0, N-i-1)$ Comparison

Iteration Comparison






```cpp
// ✅SORTINGS 01: BUBBLE SORT
#include<iostream>
#include<vector>
using namespace std;

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

    // Outer Loop
    for(int i=0; i < (N-1); i++){
        // Inner Loop
        for(int j=0; j < (N-i-1); j++){
            // When a>b where a=j and b=j+1 to swap kardo
            if(arr[j] > arr[j+1]){
                swap(arr[j], arr[j+1]);
            }
        }
    }
}

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

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

/*
INPUT: {5,4,3,1,2}
OUTPUT: {1,2,3,4,5}
TIME COMPLEXITY: O(N^2)
SPACE COMPLEXITY: O(1)
*/
```