// not able to

<https://github.com/Vaibhav-kira/cpp_codes/blob/master/cpp%20codes/array/arrayProb/FirstRepeatingValue.cpp>

<https://github.com/Vaibhav-kira/cpp_codes/blob/master/cpp%20codes/array/arrayProb/subarrayGivenSum.cpp>

<https://github.com/Vaibhav-kira/cpp_codes/tree/master/cpp%20codes/array/arrayProb>

Int min = INT\_MIN;

INT\_MIN = minimum integer possible in c++

INT\_MAX; header required is climits

#include<cllimits>

If using sizeof(arr) in function use it in for loop as arr is passed as a pointer in array

Binary search O(logn) base 2

**Selection sort**

Find the smallest element in the array and swap it with the element in the begineng;

for(int i=0;i<*n*-1;i++)

    {

        for(int j=i+1;j<*n*;j++)

        {

            int a = *arr*[j];

            int b = *arr*[i];

            if(a<b)

            {

*arr*[i] = a;

*arr*[j] = b;

            }

        }

    }

**Bubble sort**

Swap the numbers if they are in wrong order

for(int i=0;i<*n*;i++)

    {

        int flag = 0;

        for(int j=0;j<*n*-1-i;j++)

        {

            if(*arr*[j]>*arr*[j+1])

            {

                flag = 1;

                int temp = *arr*[j];

*arr*[j] = *arr*[j+1];

*arr*[j+1] = temp;

                /\*

                arr[j] = arr[j] + arr[j+1];

                arr[j+1] = arr[j] - arr[j+1];

                arr[j] = arr[j] - arr[j+1];

                \*/

            }

        }

        if(flag==0)

        {

            return;

        }

    }

**Insertion sort**

Insert an unsorted array to its’s correct position in sorted array

for(int i=0;i<*n*;i++)

    {

        if(*arr*[i+1]<*arr*[i])

        {

            for(int j=i;j>=0;j--)

            {

                if(*arr*[j+1]<*arr*[j])

                {

*arr*[j] = *arr*[j] + *arr*[j+1];

*arr*[j+1] = *arr*[j] - *arr*[j+1];

*arr*[j] = *arr*[j] - *arr*[j+1];

                }

            }

        }

    }

All are of O(n^2) complexity ….

Max till ith element is a non decreasing value i.e. when we increase i max no. increases …..

Subarray vs subsequence ?

<https://drive.google.com/drive/folders/1-PNukHSuFB4NsmGI1U4772DBXr3eOoxR>

Subarray

Subarray is a continuous part of the array.

Note: Number of subarrays of an array with n elements = n

nC2 + n = n\*(n+1) / 2.

nC2 for selecting pair i j and n as i and j can be same also.

Subsequence

A subsequence is a sequence that can be derived from an array by selecting zero

or more elements, without changing the order of the remaining elements.

Note: Number of subsequences of an array with n elements = 2^n

We have two options one to take and one note to take i.. 2\*2\*2\*….n

Ever subarray is a subsequense but every subsequebce is not a sub array.

**Sum of subarray**

for(int i=0;i<*n*;i++)

    {

        int sum = 0;

        for(int j=i;j<*n*;j++)

        {

            sum += *arr*[j];

            cout<<sum<<endl;

        }

    }

**Longest arthimetic subarray**

[**https://github.com/Vaibhav-kira/cpp\_codes/blob/master/cpp%20codes/array/arrayProb/longestArtimeticSubarray.cpp**](https://github.com/Vaibhav-kira/cpp_codes/blob/master/cpp%20codes/array/arrayProb/longestArtimeticSubarray.cpp)

**Two pointer approach**

Take two pointers/ elements in an array

<https://github.com/Vaibhav-kira/cpp_codes/blob/master/cpp%20codes/array/arrayProb/subarrayGivenSum.cpp>

**Very Important: -**

[**https://github.com/Vaibhav-kira/cpp\_codes/blob/master/cpp%20codes/array/arrayProb/maximumsubarraysum.cpp**](https://github.com/Vaibhav-kira/cpp_codes/blob/master/cpp%20codes/array/arrayProb/maximumsubarraysum.cpp)

sort() has a time complexity of O(N\*log(N))