**BINARY SEARCH**

**Keywords / Use Cases :** Maximum and Minimum Value , Search Space , We can eliminate a Particular part by finding a temporary solution

**Time Complexity : O(nlogn)**

**Space Complexity : O(1)**

**Problems :**

1. Pivot Element in an Array .
2. First and Last Position in an Array. (Leet Code 34)
3. Search in Sorted rotated Array . (Leet Code 33)
4. Square Root of a Number (Leet Code).
5. Page Partitioning (Code Studio)
6. Painters and Cows (Code Studio)
7. Aggressive Cows (Code Studio)

On 2D Arrays :

Mid = s + (e-s)/2;

Element = matrix[mid/col][mid%col]

**SELECTION SORT**

**Use Cases / Key Words :** Small Size Array .

**Time Complexity : O(n2)**

* Best Case : O(n2)
* Worst Case : O(n2)

**Space Complexity : O(1)**

**BUBBLE SORT**

**Key Words / Use Cases :** In ith Round we are placing the ith largest element in its right place.

**Time Complexity : O(n2)**

* Best Case : O(n)
* Worst Case : O(n2)

**Space Complexity : O(1)**

**INSERTION SORT**

**Keyword / Use Cases** : Adaptable method then Selection and Bubble Sort

**Time Complexity : O(n2)**

* Best Case : O(n)
* Worst Case : O(n2)

**Space Complexity : O(1)**

**STL (Standard Library Functions)**

**1.Vector** : It is adynamic array which can increase or decrease its size dynamically . When the array is full and we are trying to insert another element then the vector creates another array having size 2 times of the previous array and copies all the element from old array to the new array.

* Random access is possible.
* Direct access is possible.

**2.DeQue :** Deque (double-ended queue) is an indexed sequence container that allows fast insertion and deletion at both its beginning and its end**.**

* Random access is possible.
* Direct access is possible.

**3.List:** In C++ List DS is by default a Double Linked List.

* Random access is not possible.
* Direct ace;ss is not possible.

**4. Queue :** A queue is a data structure that is optimized for a specific access pattern: the “first in, first out” (FIFO) pattern that describes lines as we know them in everyday life

**5. Priority Queue :** A priority queue in c++ is a type of container adapter, which processes only the highest priority element, i.e. the first element will be the maximum of all elements in the queue, and elements are in decreasing order.

**->**It is 2 types : i) Max Heap

ii) Min Heap

**ARRAYS**

**Some Array Questions :**

1. Reverse the Array (Code Studio)
2. Merge Sorted Array (Leet code 88)
3. Move Zeros (Leet Code 283)
4. Rotate Array (Leet Code 189)
5. Check if Array Is Sorted and Rotated ( Leet Code 1752)

**STRINGS**

**Note :** cin.getline (string, length, 'delimeter');

**Problems Solved :**

1. Valid Palindrome (Leet Code)
2. Reverse Words in String (Leet Code 151)
3. Highest Occuring Character
4. Permutation in String (Leet Code 567)
5. Remove All Adjacent Duplicates In String (Leet Code 1047)
6. String Compression ( Leet Code 443)

**Mathematics for DSA**

**1.Count Prime Numbers between O to N. (Using Sieve Of Eratosthenes)**

**GCD & LCM Formulae:**

1. gcd(a,b) = gcd (a-b , b)
2. gcd(a,b) = gcd (a%b , b) //Use Both the formulae until one of the value becomes 0
3. lcm(a,b) \* gcd(a,b) = a\*b

**Find Power :**

If We want to find power i.e a^n we can do this by this

1. a^n = (a^n/2)^2 // If n is even
2. a^n = (a^n/2)^2 \* a //if n is Odd

Time Complexity is O(log n)

**POINTERS**

**RECURSION**

**Problems :**

1. Fibonacci Number
2. Climbing Stairs (Leet code 70)
3. Print Digits of a number (VS code)