



DSA SERIES

- Learn Coding



Topic to be Covered today

Binary Search Problems



LETS START TODAY'S LECTURE

704. Binary Search

```
class Solution {
public:
    int search(vector<int>& nums, int target) {
        int n = nums.size();

        int low = 0, high = n - 1;

        while (low <= high) {
            int mid = (low + high) / 2;
            // int mid = low + (high-low)/2;

            if (nums[mid] == target) {
                return mid;
            } else if (nums[mid] > target) {
                high = mid - 1;
            } else {
                low = mid + 1;
            }
        }
        return -1;
    }
};
```

374. Guess Number Higher or Lower

```
class Solution {
public:
    int guessNumber(int n) {
        int low = 1, high = n;

        while (low <= high) {
            int mid = low + (high-low)/2;

            int a = guess(mid);

            if (a == 0) {
                return mid;
            } else if (a == -1) {
                high = mid - 1;
            } else {
                low = mid + 1;
            }
        }
        return 0;
    }
};
```

274. H-Index

```
class Solution {
public:
    int hIndex(vector<int>& citations) {
        int n = citations.size();

        sort(rbegin(citations),rend(citations));

        int idx = 0;

        while(idx < n && citations[idx] >idx){
            idx++;
        }

        return idx;
    }
};
```

275. H-Index II

```
class Solution {
public:
    int hIndex(vector<int>& citations) {
        int n = citations.size();

        int low = 0, high = n - 1;

        while (low <= high) {
            int mid = low + (high - low) / 2;

            if (citations[mid] >= n - mid) {
                high = mid - 1;
            } else {
                low = mid + 1;
            }
        }

        return n - low;
    }
};
```

34. Find First and Last Position of Element in Sorted Array

```
class Solution {
public:
    vector<int> searchRange(vector<int>& nums, int target) {

        int n = nums.size();
        int low = 0, high = n-1;

        int f = -1 , l = -1;

        while( low <= high){
            int mid = low + (high-low)/2;

            if(nums[mid] == target){
                f = mid;
                high = mid-1;
            } else if(nums[mid] <target){
                low = mid+1;
            } else {
                high = mid-1;
            }
        }
    }
}
```



```
low = 0;
high = n-1;

while(low<= high){
    int mid = low + (high-low)/2;

    if(nums[mid] == target){
        l = mid;
        low = mid+1;
    } else if(nums[mid]<target){
        low = mid+1;
    } else {
        high = mid-1;
    }
}
return {f,l};
}
};
```

1608. Special Array With X Elements Greater Than or Equal X

```
class Solution {
public:
    int specialArray(vector<int>& nums) {
        int low = 0;
        int high = nums.size();

        while (low <= high) {
            int mid = (low + high) / 2;

            int count = 0;

            for (int num : nums) {
                if (num >= mid) {
                    count++;
                }
            }
            if (count == mid) {
                return mid;
            } else if (count > mid) {
                low = mid + 1;
            }
        }
    }
};
```

```
        } else {  
            high = mid - 1;  
        }  
    }  
    return -1;  
}  
};
```



Learn coding

THANK YOU