



DSA SERIES

- Learn Coding



Topic to be Covered today

Heap Questions



LETS START TODAY'S LECTURE

215. Kth Largest Element in an Array

```
class Solution {
public:
    int findKthLargest(vector<int>& nums, int k) {
        priority_queue<int> que(nums.begin(), nums.end());

        int count = 1;

        while (que.size()) {
            if (count == k)
                return que.top();

            count++;
            que.pop();
        }
        return 0;
    }
};
```

347. Top K Frequent Elements

```
class Solution {
public:
    vector<int> topKFrequent(vector<int>& nums, int k) {
        unordered_map<int,int> mp;

        for(int &num : nums){
            mp[num]++;
        }

        priority_queue<pair<int,int>> que;

        for(auto it : mp){
            que.push({it.second, it.first});
        }

        vector<int> result ;
```

```
while(k--){
    auto it = que.top();
    que.pop();

    result.push_back(it.second);
}

return result;
};
```

1845. Seat Reservation Manager

```
class SeatManager {
public:
    int seat;
    priority_queue<int, vector<int>, greater<int>> pq;
    SeatManager(int n) { seat = 1; }

    int reserve() {
        if (!pq.empty()) {
            int mark = pq.top();
            pq.pop();

            return mark;
        }

        int mark = seat;
        seat++;
        return mark;
    }

    void unreserve(int seatNumber) { pq.push(seatNumber); }
};
```

451. Sort Characters By Frequency

```
class Solution {
public:
    string frequencySort(string s) {
        unordered_map<char,int> mp;

        for(char ch : s){
            mp[ch]++;
        }

        priority_queue<pair<int,int>> pq;

        for(auto it : mp){
            pq.push({it.second, it.first});
        }

        string result = "";
```



```
while(!pq.empty()){  
    auto it = pq.top();  
    pq.pop();  
  
    for(int i = 0;i<it.first;i++){  
        result += it.second;  
    }  
}
```

```
return result;
```

```
};  
}
```

1054. Distant Barcodes

```
class Solution {
public:
    vector<int> rearrangeBarcodes(vector<int>& barcodes) {
        unordered_map<int, int> mp;
        int n = barcodes.size();

        for (int& num : barcodes) {
            mp[num]++;
        }

        priority_queue<pair<int, int>> que;

        for (auto it : mp) {
            que.push({it.second, it.first});
        }

        vector<int> result(n);
        int count = 0;
```

```
while (!que.empty()) {
    auto it = que.top();
    que.pop();

    int freq = it.first;

    while (freq-- > 0) {
        if (count >= n) {
            count = 1;
        }
        result[count] = it.second;
        count += 2;
    }
}
return result;
};
```



Learn coding

THANK YOU