241. Different Ways to Add Parentheses

single-if-multi-option faster than multi-if-single-option

347. Top K Frequent Elements

Top-Frequecne Question

These questions has samiliar solutions.

Solution

HashMap<num, frequence> to calc frequence(O(n)), then use **bucket sort** (O(n)) or **save keys in vector then sort by frequence**(O(nlogn)).

Or just make a **max Heap which size is (n - k)**, all of the top-k value will be poped out.

How to sort sequence by params from other struct?

Define a hashmap previously : unordered_map<int, int> `m` , vector `keys` is keys in hashmap;

Then use a lambda expression in sort:

```
//A lambda expression in function sort's compare part:
sort(keys.begin(), keys.end(), [&m](int a, int b){
    return m[a] > m[b];
});
```

406. Queue Reconstruction by Height

How to deal with "equal to" condition in sort?

Usually we sort a sequence by one of the struct's param, but in some case, it can be difficult to deal with the "equal to" condition. How can I **sort the "equal to" part sequence by struct's some other params**?

Solution

That sort will consider `first` param firstly, and if a.first equal to b.first, it will compared by `second` param.

```
sort(seq.begin(), seq.end(), [](pair<int, int> a, pair<int, int> b){
    return a.first == b.first ? a.second < b.second : a.first > b.first;
});
```

744. Find Smallest Letter Greater Than Target

Problem

while l = r - 1 , l + (r - l) / 2 will **always equal to l** , so can not quit correctly in find most-right location task.

example

find:

l = 0, r = 1

n[l] = 3, n[r] = 4, target = 3

m = 0 + (1 - 0) / 2 = 0

n[m]<=n[l]

l = m

find:

l = m = 0, r= 1

....

Solution

- in most-left location task, we use m = l + (r - l) / 2
- in most-right location task, we use m = l + (r - l + 1) /2