502. IPO

Add to List

Description SubmissionsSolutions

- Total Accepted: **2097**
- Total Submissions: **6288**
- Difficulty: **Hard**
- Contributors: **μsic_forever**

Suppose LeetCode will start its IPO soon. In order to sell a good price of its shares to Venture Capital, LeetCode would like to work on some projects to increase its capital before the IPO. Since it has limited resources, it can only finish at most **k** distinct projects before the IPO. Help LeetCode design the best way to maximize its total capital after finishing at most **k** distinct projects.

You are given several projects. For each project **i**, it has a pure profit $\mathbf{P}_i$ and a minimum capital of $\mathbf{C}_i$ is needed to start the corresponding project. Initially, you have **W** capital. When you finish a project, you will obtain its pure profit and the profit will be added to your total capital.

To sum up, pick a list of at most **k** distinct projects from given projects to maximize your final capital, and output your final maximized capital.

Example 1:

```
Input: k=2, W=0, Profits=[1,2,3], Capital=[0,1,1].


Output: 4


Explanation: Since your initial capital is 0, you can only start the project
 indexed 0.
             After finishing it you will obtain profit 1 and your capital bec
omes 1.
```

```
            With capital 1, you can either start the project indexed 1 or th
e project indexed 2.

            Since you can choose at most 2 projects, you need to finish the
project indexed 2 to get the maximum capital.

            Therefore, output the final maximized capital, which is 0 + 1 +
3 = 4.
```

Note:

1. You may assume all numbers in the input are non-negative integers.
2. The length of Profits array and Capital array will not exceed 50,000.
3. The answer is guaranteed to fit in a 32-bit signed integer.

Subscribe to see which companies asked this question.

```
class Solution {
public:
    int findMaximizedCapital(int k, int W, vector<int>& Profits,
vector<int>& Capital) {
        priority_queue<int> low;
        multiset<pair<int,int>> high;
        for(int i=0;i<Profits.size();i++)
        {
            if(Profits[i]>0){
                if(Capital[i]<=W) low.push(Profits[i]);
                else high.emplace(Capital[i],Profits[i]);
            }


        }
        while(k-- && low.size())
        {
            W+=low.top(); low.pop();
            //multiset<pair<int,int>>::iterator it;
            for(auto it=high.begin();it->first<=W && high.size();it =
high.erase(it))
            {
                low.push(it->second);
```

```
            }
        }
        return W;
    }
};
```