August 13, 2018 题目：459,486,451,407,218,422,831,177,79,715,804,262,174,197

459

1. $O(\sqrt{N})$ Brute Force:

```
class Solution {
    bool rep(string &s, int i){
        if(i == s.size()) return false;
        for(int j=i; j<s.size(); j+=i) if(s.substr(j, i) != s.substr(0, i)) return false;
        return true;
    }
public:
    bool repeatedSubstringPattern(string s) {
        int n = s.size();
        for(int i=1; i<=int(sqrt(n)); ++i) if(n%i==0){
            if(rep(s, i) || rep(s, n/i)) return true;
        }
        return false;
    }
};
```

我回来刷题了：basically，字符串的第一个char是repeating string的第一个char，最后一个char是repeating string的最后一个char，令S1 = S+ S(S is the input string),掐头去尾，如果S还在S1 的话，return true else false

```
class Solution:
    def repeatedSubstringPattern(self, s):
        """
        :type s: str
        :rtype: bool
```

```
        """
        St = s + s
        return St[1:-1].find(s) != -1
```

486

1. 2D dp:

```
class Solution {
public:
    bool PredictTheWinner(vector<int>& nums) {
        int n = nums.size();
        vector<int> P(n+1, 0);
        vector<vector<int>> dp(n, vector<int>(n, 0));
        for(int i=0; i<n; ++i) {
            dp[i][i] = nums[i];
            P[i+1] = P[i] + nums[i];
        }
        for(int l=1; l<n; ++l) for(int i=0; i+l<n; ++i){
            int j = i+l;
            dp[i][j] = max(nums[i] + P[j+1]-P[i+1]-dp[i+1][j],
nums[j] + P[j] - P[i] - dp[i][j-1]);
        }
        return 2 * dp[0][n-1] >= P[n];
    }
};
```

451

1. `O(log(n))`:

```
class Solution {
    typedef pair<int, int> ii;
public:
    string frequencySort(string s) {
```

```
        unordered_map<char, int> cnt;
        for(char c: s) ++cnt[c];
        set<ii, greater<ii>> res;
        for(int i=0; i<s.size(); ++i) if(cnt.count(s[i])){
            res.insert(ii(cnt[s[i]], -i));
            cnt.erase(s[i]);
        }
        string ans;
        for(auto p: res) ans += string(p.first, s[-p.second]);
        return ans;
    }
};
```

2. `O(n)` Bucket sort, 但没快多少：

```
class Solution {
public:
    string frequencySort(string s) {
        unordered_map<char, int> cnt;
        for(char c: s) ++cnt[c];
        vector<string> buc(s.size());
        for(auto p: cnt) buc[p.second-1].append(p.second, p.fi
rst);
        string ans;
        for(int i=s.size()-1; i>=0; --i) ans.append(buc[i]);
        return ans;
    }
};
```

407

1. Maintain frontier: 做过

```
class Solution {
    typedef pair<int, int> ii;
```

```
    int dx[4] = {1, -1, 0, 0};
public:
    int trapRainWater(vector<vector<int>>& H) {
        if(H.size()<=2 || H[0].size()<=2) return 0;
        priority_queue<ii, vector<ii>, greater<ii>> Q;
        int n = H.size(), m = H[0].size(), ans = 0;
        vector<bool> U(n*m, false);
        for(int i=0; i<n; ++i){
            U[i*m] = U[i*m + m-1] = true;
            Q.push(ii(H[i][0], i*m));
            Q.push(ii(H[i][m-1], i*m + m - 1));
        }
        for(int j=1; j<m-1; ++j){
            U[j] = U[(n-1)*m + j] = true;
            Q.push(ii(H[0][j], j));
            Q.push(ii(H[n-1][j], (n-1)*m + j));
        }
        while(!Q.empty()){
            int h = Q.top().first, x = Q.top().second/m, y =
Q.top().second%m;
            Q.pop();
            for(int k=0; k<4; ++k){
                int i = x + dx[k], j = y + dx[3-k];
                if(i>=0 && i<n && j>=0 && j<m && !U[i*m+j]){
                    U[i*m + j] = true;
                    ans += max(0, h-H[i][j]);
                    Q.push(ii(max(H[i][j], h), i*m+j));
                }
            }
        }
```

```
        return ans;
    }
};
```

218

1. Process every possible key points: (做麻烦了，所以不是很快，直接heap更快，应该)

```
class Solution {
public:
    vector<pair<int, int>> getSkyline(vector<vector<int>>& bui
ldings) {
        unordered_map<int, vector<int>> B, E;
        set<int> K;
        map<int, int, greater<int>> S;
        vector<pair<int, int>> ans;
        for(auto vec: buildings){
            K.insert(vec[0]);
            K.insert(vec[1]);
            B[vec[0]].push_back(vec[2]);
            E[vec[1]].push_back(vec[2]);
        }
        int cur = 0;
        for(int k: K){
            if(B.count(k)) for(int h: B[k]) ++S[h];
            if(E.count(k)) for(int h: E[k]) {
                --S[h];
                if(!S[h]) S.erase(h);
            }
            int tmp = S.begin()->first;
            if(tmp!=cur){
                cur = tmp;
                ans.push_back(pair<int, int>(k, cur));
```

```
            }
        }
        return ans;
    }
};
```

- [ ] https://briangordon.github.io/2014/08/the-skyline-problem.html @Zebo L

422

1. Use Concept `O(n)`:

```
class Solution {
public:
    bool validWordSquare(vector<string>& W) {
        if(W.empty() || W[0].empty()) return false;
        int n = W.size();
        for(int i=0; i<n; ++i) {
            int m = W[i].size();
            if(m > n) return false;
            for(int j=0; j<m; ++j){
                if(W[j].size()<= i || W[i][j] != W[j][i]) retu
rn false;
            }
        }
        return true;
    }
};
```

831

1. 除了题目长：

```
class Solution(object):
    _ref = "0123456789 ()+-"
    def maskPII(self, S):
```

```
        if S[0] in self._ref:
            digs = ''.join([c for c in S if c in self._ref[:1
0]])
            res = "***-***-" + digs[-4: ]
            if len(digs) > 10:
                res = "+" + (len(digs) - 10) * "*" + "-" + res
                if S[0] == '-':
                    res[0] = '-'
            return res
        return (S[0] + 5 * "*" + S.split("@")[0][-1] + "@" +
S.split("@")[-1]).lower()
```

177

SQL

79

1. 一个dfs就这么过了：

```
class Solution {
    typedef vector<bool> vb;
    int n, m, d[4]={1, -1, 0, 0};
    bool dfs(int k, string &s, vector<vector<char>>& B, int i,
int j, vector<vb> &pass){
        if(k == s.size()) return true;
        if(i<0 || i>=B.size() || j<0 || j>=B[0].size() || s
[k]!=B[i][j] || pass[i][j]) return false;
        pass[i][j] = true;
        for(int t=0; t<4; ++t) if(dfs(k+1, s, B, i+d[t], j+d[3
-t], pass)) return true;
        return pass[i][j] = false;
    }
public:
    bool exist(vector<vector<char>>& B, string word) {
```

```
        if(B.empty() || B[0].empty()) return false;
        vector<vb> pass(B.size(), vb(B[0].size(), false));
        for(int i=0; i<B.size(); ++i) for(int j=0; j<B[0].size
(); ++j) if(dfs(0, word, B, i, j, pass)) return true;
        return false;
    }
};
```

715

1. The same as interval processing problems:

```
class RangeModule {
    const int inf = int(1E9) + 7;
    map<int, int> R;
public:
    RangeModule() {
        R[-inf] = -inf;
        R[inf] = inf;
    }
    void addRange(int left, int right) {
        auto it = --R.upper_bound(left);
        if(it->second<left) ++it;
        left = min(left, it->first);
        while(it!=R.end() && it->first<=right){
            right = max(right, it->second);
            it = R.erase(it);
        }
        R[left] = right;
    }
    bool queryRange(int left, int right) {
        auto it = --R.upper_bound(left);
        return it->second >= right;
```

```
    }
    void removeRange(int left, int right) {
        auto it = --R.lower_bound(left);
        if(it->second > right){
            int e = it->second;
            it->second = min(it->second, left);
            R[right] = e;
            return;
        }
        it->second = min(it->second, left);
        ++it;
        while(it!=R.end() && it->second <= right) it = R.erase
(it);
        if(it->first < right){
            int e = it->second;
            R.erase(it);
            R[right] = e;
        }
    }
};
```

804

1. Short is Beauty:

```
class Solution(object):
    _ref = [".-","-...","-.-.","-..",".","..-.","-
-.","....","..",".---","-.-",".-..","--","-.","---",".--.","-
-.-",".-.","...","-","..-","...-",".--","-..-","-.--","--.."]
    def uniqueMorseRepresentations(self, words):
        return len(set(['''.join([self._ref[ord(c)-ord('a')] fo
r c in s]) for s in words]))
```

262

Hard SQL

174

1. How can this be a hard:

```
class Solution {
public:
    int calculateMinimumHP(vector<vector<int>>& D) {
        if(D.empty() || D[0].empty()) return 1;
        int n = D.size(), m = D[0].size();
        vector<int> dp(m, 1);
        for(int j=m-2; j>=0; --j) dp[j] = max(1, dp[j+1] - D[n
-1][j+1]);
        for(int i=n-2; i>=0; --i){
            dp[m-1] = max(1, dp[m-1] - D[i+1][m-1]);
            for(int j=m-2; j>=0; --j){
                dp[j] = min(max(1, dp[j+1] - D[i][j+1]), max
(1, dp[j] - D[i+1][j]));
            }
        }
        return max(1, dp[0] - D[0][0]);
    }
};
```

197

SQL again !!!
Rip haven’t learnt SQL yet 😣