September 10, 2018 题目：803,318,849,249,447,495,136,743,342,443

803

1. Union find:

```
class Solution {
    vector<int> P;
    unordered_map<int, int> area;
    int findRoot(int i){
        assert(P[i] >= 0);
        if(P[i] == i) return i;
        return P[i] = findRoot(P[i]);
    }
    int getArea(int i){
        return area[findRoot(i)];
    }
    int dr[4] = {1, -1, 0, 0};
    void merge(int i, int j){
        int x = findRoot(i);
        int y = findRoot(j);
        if(x == y) return;
        P[y] = x;
        area[x] += area[y];
    }
public:
    vector<int> hitBricks(vector<vector<int>>& grid, vector<vector<int>>& hits) {
        int n = grid.size(), m = grid[0].size();
```

```
        vector<int> Q;
        unordered_set<int> S;
        for(int i=hits.size()-1; i>=0; --i) {
            Q.push_back(hits[i][0] * m + hits[i][1]);
            S.insert(hits[i][0] * m + hits[i][1]);
        }
        P.resize(n*m + 1, -1);
        P[n*m] = n*m;
        for(int j=0; j<m; ++j) if(grid[0][j] && !S.count(j)) {
            P[j] = n*m;
            ++area[n*m];
        }
        for(int i=1; i<n; ++i) for(int j=0; j<m; ++j) if(grid
[i][j] && !S.count(i*m + j)) {
            P[i*m + j] = i*m + j;
            area[i*m + j] = 1;
            for(int k=0; k<4; ++k){
                int x = i+dr[k], y = j+dr[3-k];
                if(x>=0 && x<n && y>=0 && y<m && P[x*m + y] >=
0) merge(i*m + j, x*m + y);
            }
        }
        vector<int> res{area[findRoot(n*m)]}, ans;
        for(int z: Q){
            int i = z/m, j = z%m;
            if(!grid[i][j]) {
                res.push_back(res.back());
                continue;
            }
            P[z] = z;
```

```
            area[z] = 1;
            if(!i) merge(n*m, z);
            for(int k=0; k<4; ++k){
                int x = i+dr[k], y = j+dr[3-k];
                if(x>=0 && x<n && y>=0 && y<m && P[x*m + y] >=
0) merge(z, x*m + y);
            }
            res.push_back(area[findRoot(n*m)]);
        }
        for(int i=res.size()-1; i>0; --i) ans.push_back(max(0,
res[i] - res[i-1]-1));
        return ans;
    }
};
```

318

1. Bit map:

```
class Solution {
    #define CI(c) int((c) - 'a')
public:
    int maxProduct(vector<string>& words) {
        unordered_map<int, int> L;
        for(string &s: words){
            int tmp = 0;
            for(char &c: s) tmp |= (1<<CI(c));
            L[tmp] = max(L[tmp], (int)s.size());
        }
        int ans = 0;
        for(auto it=L.begin(); it!=L.end(); ++it) for(auto ir=
it; ir!=L.end(); ++ir) if(!(it->first & ir->first)){
            ans = max(ans, it->second * ir->second);
```

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

849

1. 注意下边界即可：

```
class Solution {
public:
    int maxDistToClosest(vector<int>& seats) {
        int ans = 0, cnt = 0, i = 0;
        while(i<seats.size() && !seats[i]) ++i;
        ans = i;
        for(auto n: seats){
            if(n) {
                ans = max(ans, (cnt-1)/2+1);
                cnt = 0;
            }
            else ++cnt;
        }
        return max(ans, cnt);
    }
};
```

249

1. 做hash：

```
class Solution {
    #define MOVE(c, c0) char(int((c) - (c0) + 26)%26 + 'a')
public:
    vector<vector<string>> groupStrings(vector<string>& string
s) {
```

```
        unordered_map<string, vector<string>> D;
        for(string s: strings){
            if(s.empty()) D[s].push_back(s);
            else{
                string tmp = s;
                for(char &c: tmp) c = MOVE(c, s[0]);
                D[tmp].push_back(s);
            }
        }
        vector<vector<string>> ans;
        for(auto p: D) ans.push_back(p.second);
        return ans;
    }
};
```

447

1. 做过，很简单：

```
class Solution {
public:
    int numberOfBoomerangs(vector<pair<int, int>>& points) {
        int ans = 0;
        for(int i=0;i<points.size();++i){
            unordered_map<int, int> cnt;
            for(int j=0; j<points.size(); ++j) if(j!=i) {
                int dis = (points[j].first - points[i].first)
* (points[j].first - points[i].first);
                dis += (points[j].second - points[i].second)*
(points[j].second - points[i].second);
                cnt[dis]++;
            }
            for(auto p: cnt) ans += p.second * (p.second - 1);
```

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

495

1. Short is Beauty 系列 :

```
class Solution(object):
    def findPoisonedDuration(self, timeSeries, duration):
        return sum([min(timeSeries[i+1] - timeSeries[i], durat
ion) for i in range(len(timeSeries)-1)]) + bool(len(timeSerie
s)) * duration
```

136

1. Short is Beauty 系列 :

```
class Solution(object):
    def singleNumber(self, nums):
        return reduce(lambda x, y: x^y, nums)
```

743

1. BFS:

```
class Solution {
    vector<unordered_map<int, int>> E;
public:
    int networkDelayTime(vector<vector<int>>& times, int N, in
t K) {
        E.resize(N+1);
        vector<int> res(N+1, INT_MAX);
        res[K] = 0;
        for(auto v: times) E[v[0]][v[1]] = v[2];
        queue<int> Q;
        Q.push(K);
```

```
        while(!Q.empty()){
            int k = Q.front();
            Q.pop();
            for(auto p: E[k]) if(res[k] + p.second < res[p.fir
st]){
                res[p.first] = res[k] + p.second;
                Q.push(p.first);
            }
        }
        int ans = 0;
        for(int i=1; i<=N; ++i) ans = max(ans, res[i]);
        return (ans==INT_MAX? -1: ans);
    }
};
```

2. Dijkstra 思想，还没我的快：

```
class Solution {
    typedef pair<int, int> ii;
    typedef vector<int> vi;
    vector<vector<ii>> E;
public:
    int networkDelayTime(vector<vector<int>>& times, int N, in
t K) {
        E.resize(N);
        int res = 0;
        unordered_set<int> used;
        for(vi v: times) E[v[0]-1].push_back(ii(v[1]-1, v
[2]));
        priority_queue<ii, vector<ii>, greater<ii>> Q;
        Q.push(ii(0, K-1));
        while(!Q.empty()){
            int k = Q.top().second, dis = Q.top().first;
```

```
            Q.pop();
            if(!used.count(k)){
                res = max(res, dis);
                used.insert(k);
                for(auto p: E[k]) if(!used.count(p.first)) {
                    Q.push(ii(p.second + dis, p.first));
                }
            }
        }
        return (used.size() == N? res: -1);
    }
};
```

342

1. Short is Beauty 系列：

```
import math
class Solution(object):
    def isPowerOfFour(self, num):
        return num > 0 and int(4**int(math.log(num)/math.log
(4))) == num
```

443

1. 细心点，现在做题好毛糙：

```
class Solution {
public:
    int compress(vector<char>& chars) {
        int ans = 0;
        for(int i=1, cnt=1; i<=chars.size(); ++i){
            if(i<chars.size() && chars[i]==chars[i-1]) ++cnt;
            else{
                chars[ans++] = chars[i-1];
```

```
                if(cnt > 1) for(char c: to_string(cnt)) chars
[ans++] = c;
                cnt = 1;
            }
        }
        return ans;
    }
};
```