July 25, 2018 题目：679,295,684,286,769,500,428,277,228,146,702,635

679

1. 暴力枚举

```
class Solution {
    const double delta = 1E-7;
    typedef vector<double> vd;
    bool F(vd& A){
        if(A.size() == 1) return abs(A[0]-24) < delta;
        int n = A.size();
        for(int i=0; i<n; ++i) for(int j=0; j<n; ++j) if(i!=j)
{
            vd tmp;
            for(int k=0; k<n; ++k) if(k!=i && k!=j) tmp.push_b
ack(A[k]);
            double x = A[i], y = A[j];
            tmp.push_back(x + y);
            if(F(tmp)) return true;
            tmp.pop_back();
            tmp.push_back(x - y);
            if(F(tmp)) return true;
            tmp.pop_back();
            tmp.push_back(x * y);
            if(F(tmp)) return true;
            tmp.pop_back();
            if(abs(y)>delta){
                tmp.push_back(x/y);
```

```
                if(F(tmp)) return true;
                tmp.pop_back();
            }
        }
        return false;
    }
public:
    bool judgePoint24(vector<int>& nums) {
        vd ans;
        for(int k: nums) ans.push_back(double(k));
        return F(ans);
    }
};
```

2. 同上

```
class Solution {
    public boolean judgePoint24(int[] nums) {
        double[] input = new double[nums.length];
        for (int i = 0; i < nums.length; i++) {
            input[i] = (double) nums[i];
        }
        return dfs(input, input.length);
    }

    public boolean dfs(double[] input, int num) {
        if (num == 1) {
            return Math.abs(input[0] - 24) < 1e-7;
        }

        for (int i = 0; i < num; i++) {
            for (int j = i + 1; j < num; j++) {
```

```
                double first = input[i];
                double second = input[j];

                input[j] = input[num - 1];

                input[i] = first + second;
                if (dfs(input, num - 1)) {
                    return true;
                }

                input[i] = first - second;
                if (dfs(input, num - 1)) {
                    return true;
                }

                input[i] = second - first;
                if (dfs(input, num - 1)) {
                    return true;
                }

                input[i] = second * first;
                if (dfs(input, num - 1)) {
                    return true;
                }

                if (second != 0) {
                    input[i] = first / second;
                    if (dfs(input, num - 1)) {
                        return true;
                    }
```

```
                }

                if (first != 0) {
                    input[i] = second / first;
                    if (dfs(input, num - 1)) {
                        return true;
                    }
                }

                input[i] = first;
                input[j] = second;
            }
        }
        return false;
    }
}
```

295

1. 两个对位的 `priority_queue`

```
class MedianFinder {
public:
    /** initialize your data structure here. */
    priority_queue<int, vector<int>, greater<int>> U;
    priority_queue<int, vector<int>, less<int>> D;
    MedianFinder() {

    }

    void addNum(int num) {
        if(U.empty() || D.empty()) {
```

```
            U.push(num);
            if(U.size() > D.size() + 1){
                D.push(U.top());
                U.pop();
            }
        }
        else{
            if(num >= D.top()) {
                U.push(num);
                if(U.size() > D.size() + 1){
                    D.push(U.top());
                    U.pop();
                }
            }
            else{
                D.push(num);
                if(D.size() > U.size()){
                    U.push(D.top());
                    D.pop();
                }
            }
        }
    }

    double findMedian() {
        if(U.size() > D.size()) return double(U.top());
        else return 0.5 * (U.top() + D.top());
    }
};
```

2.

```
class MedianFinder {

    /** initialize your data structure here. */
    PriorityQueue<Integer> max;
    PriorityQueue<Integer> min;
    public MedianFinder() {
        max = new PriorityQueue<>(1, Collections.reverseOrder
());
        min = new PriorityQueue<>();
    }

    public void addNum(int num) {
        max.offer(num);
        min.offer(max.poll());
        if (max.size() < min.size()) {
            max.offer(min.poll());
        }
    }

    public double findMedian() {
        return max.size() == min.size() ? (max.peek() + min.pe
ek()) / 2.0 : max.peek();
    }
}
```

684

1. Union Find

```
class Solution {
    Map<Integer, Integer> map;
    public int find(int i) {
        int parent = i;
```

```
        while (map.get(parent) != parent) {
            parent = map.get(parent);
        }
        return parent;
    }
    public int[] findRedundantConnection(int[][] edges) {
        map = new HashMap<>();
        for (int i = 1; i <= edges.length; i++) {
            map.put(i, i);
        }
        for (int[] edge : edges) {
            int i = find(edge[0]);
            int j = find(edge[1]);
            if (i != j) map.put(i, j);
            else return edge;
        }
        return new int[]{};

    }
}
```

2. 同上：

```
class Solution {
    vector<int> P;
    int findRoot(int i){
        if(P[i] == i) return i;
        return P[i] = findRoot(P[i]);
    }
public:
    vector<int> findRedundantConnection(vector<vector<int>>& e
dges) {
```

```
        int n = edges.size();
        P.resize(n+1);
        for(int i=1; i<=n; ++i) P[i] = i;
        for(auto e: edges){
            if(findRoot(e[0]) == findRoot(e[1])) return vector
<int>{min(e[1], e[0]), max(e[1], e[0])};
            P[findRoot(e[1])] = findRoot(e[0]);
        }
        return vector<int>{-1, -1};
    }
};
```

286

1. BFS 之前做过：

```
class Solution {
    const int INF = (1<<31) - 1;
    int dx[4] = {0, 0, 1, -1}, dy[4] = {1, -1, 0, 0};
public:
    void wallsAndGates(vector<vector<int>>& rooms) {
        if(rooms.empty() || rooms[0].empty()) return;
        int n = rooms.size(), m = rooms[0].size();
        queue<int> Q;
        for(int i=0;i<n;++i) for(int j=0;j<m;++j) if(!rooms[i]
[j]) Q.push(i*m + j);
        while(!Q.empty()){
            int x = Q.front()/m, y = Q.front()%m;
            Q.pop();
            for(int k=0;k<4;++k){
                int x1 = x + dx[k], y1 = y + dy[k];
                if(x1>=0 && x1<n && y1>=0 && y1<m && rooms[x1]
[y1]==INF){
```

```
                    rooms[x1][y1] = rooms[x][y] + 1;
                    Q.push(x1 * m + y1);
                }
            }
        }
        return ;
    }
};
```

769 (之前有过了7/16/2018)

+July 16, 2018

500

1. 记下每个字母所在行即可：

```
class Solution {
    vector<string> kb = {
        "qwertyuiopQWERTYUIOP",
        "asdfghjklASDFGHJKL",
        "zxcvbnmZXCVBNM"
    };
public:
    vector<string> findWords(vector<string>& words) {
        map<char, int> pos;
        for(int i=0; i<3; ++i) for(char c: kb[i]) pos[c] = i;
        vector<string> ans;
        for(string s: words){
            bool ok = true;
            for(int i=1; i<s.size() && ok; ++i) if(pos[s[i]] !
= pos[s[0]]) ok = false;
            if(ok) ans.push_back(s);
        }
```

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

428

1. Description 里面的说明给错了 正确的是 `[1 [[3[[5] [6]] [2] [4]]`, 可以直接用递归

```
class Codec {
public:

    // Encodes a tree to a single string.
    string serialize(Node* root) {
        if(!root) return "";
        if(root->children.empty()) return "[" + to_string(root
->val) + "]";
        string ans = "[" + to_string(root->val) + "[";
        for(Node *ch: root->children) ans += serialize(ch) + "
";
        ans.pop_back();
        ans += "]]";
        return ans;
    }

    // Decodes your encoded data to tree.
    Node* deserialize(string data) {
        if(data.empty()) return NULL;
        //cout << data << endl;
        data = data.substr(1, data.size()-2);
        Node *root = new Node(stoi(data));
        auto i = data.find_first_of("[") + 1;
        data = data.substr(0, data.size()-1);
```

```
        while(i<data.size() && i){
            int k = i+1, cnt = 1;
            while(k<data.size() && cnt){
                if(data[k]=='[') ++cnt;
                if(data[k]==']') --cnt;
                ++k;
            }
            root->children.push_back(deserialize(data.substr
(i, k-i)));
            i = k+1;
        }
        return root;
    }
};
```

277

1.

```
public class Solution extends Relation {
    public int findCelebrity(int n) {
        int cel = 0;
        for (int i = 1; i < n; i++) {
            if (knows(cel, i)) cel = i;
        }
        for (int i = 0; i < n; i++) {
            if (i != cel && (!knows(i, cel) || knows(cel, i)))
return -1;
        }
        return cel;
    }
}
```

2. 暴力枚举：过了

```
bool knows(int a, int b);

class Solution {
public:
    int findCelebrity(int n) {
        vector<vector<int>> dp(n, vector<int>(n, -1));
        for(int i=0; i<n; ++i){
            bool ok = true;
            for(int j=0; j<n && ok; ++j) if(j!=i){
                bool j2i, i2j;
                if(dp[i][j] < 0) dp[i][j] = int(knows(i, j));
                if(dp[j][i] < 0) dp[j][i] = int(knows(j, i));
                if(dp[i][j] || !dp[j][i]) ok = false;
            }
            if(ok) return i;
        }
        return -1;
    }
};
```

3. 模仿1 的做法：做了一些改进。2中做法简直弱爆了

```
bool knows(int a, int b);

class Solution {
    typedef pair<int, int> ii;
public:
    int findCelebrity(int n) {
        int k = 0, j = -1;
        map<ii, bool> dp;
        for(int i=1; i<n; ++i){
            if(knows(k, i)){
```

```
                j = k;
                k = i;
            }
        }
        for(int i=0; i<k; ++i) if(knows(k, i)) return -1;
        for(int i=0; i<n; ++i) if(i!=j && i!=k && !knows(i,
k)) return -1;
        return k;
    }
};
```

228

1. One pass, 无难度

```
class Solution {
public:
    vector<string> summaryRanges(vector<int>& nums) {
        vector<string> ans;
        if(nums.empty()) return ans;
        int start = nums[0], end = nums[0];
        for(int i=1; i<nums.size(); ++i){
            if(nums[i]==nums[i-1]+1) ++end;
            else{
                if(start == end) ans.push_back(to_string(star
t));
                else ans.push_back(to_string(start) + "->" + t
o_string(end));
                start = end = nums[i];
            }
        }
        if(start == end) ans.push_back(to_string(start));
```

```
        else ans.push_back(to_string(start) + "->" + to_string
(end));
        return ans;
    }
};
```

146

1. 用STL自带的list：

```
class LRUCache {
    typedef pair<int, int> ii;
    int cap;
    list<ii> L;
    unordered_map<int, list<ii>::iterator> M;
public:
    LRUCache(int capacity): cap(capacity) {}
    int get(int key) {
        if(!M.count(key)) return -1;
        cout << "Shaocong" << L.size() << endl;
        int val = M[key]->second;
        L.erase(M[key]);
        L.insert(L.begin(), ii(key, val));
        M[key] = L.begin();
        return val;
    }
    void put(int key, int value) {
        if(M.count(key)){
            L.erase(M[key]);
            L.insert(L.begin(), ii(key, value));
            M[key] = L.begin();
        }
        else{
```

```
            L.insert(L.begin(), ii(key, value));
            M[key] = L.begin();
            if(M.size() > cap){
                M.erase(L.back().first);
                L.pop_back();
            }
        }

    }
};
```

702

1.

```
class Solution {
    public int search(ArrayReader reader, int target) {
        int j = 1;
        while (reader.get(j) < target) {
            j = j << 1;
        }
        int i = j >> 1;
        while (i <= j) {
            int mid = i + (j - i) / 2;
            if (reader.get(mid) > target) {
                j = mid - 1;
            } else if (reader.get(mid) < target) {
                i = mid + 1;
            } else {
                return mid;
            }
        }
```

```
        return -1;
    }
}
```

2. 2 * Bisection search

```
class ArrayReader;

class Solution {
public:
    int search(const ArrayReader& reader, int target) {
        int l = -1, r = 20000;
        while(l < r-1){
            int c = (l+r) / 2;
            if(reader.get(c) == 2147483647) r = c;
            else l = c;
        }
        l = -1;
        while(l < r-1){
            int c = (l+r)/2;
            if(reader.get(c) > target) r = c;
            else l = c;
        }
        if(l==-1 || reader.get(l)!=target) return -1;
        return l;
    }
};
```

635

1.

```
class LogSystem {
    Map<String, Integer> map;
    public LogSystem() {
```

```
        map = new HashMap<>();
    }

    public void put(int id, String timestamp) {
        map.put(timestamp, id);
    }

    private String[] parseKey(String timestamp) {
        String[] timestamps = timestamp.split(":");
        return timestamps;
    }

    public List<Integer> retrieve(String s, String e, String g
ra) {
        int i = getIndex(gra);
        String[] keyS = parseKey(s);
        String[] keyE = parseKey(e);
        long kS = renderKey(keyS, i);
        long kE = renderKey(keyE, i);
        List<Integer> list = new ArrayList<>();
        for (String t : map.keySet()) {
            long key = renderKey(parseKey(t), i);
            if (key >= kS && key <= kE) {
                list.add(map.get(t));
            }
        }
        return list;
    }

    private long renderKey(String[] ts, int i) {
```

```
        StringBuilder sb = new StringBuilder();
        for (int j = 0; j <= i; j++) {
            sb.append(ts[j]);
        }
        return Long.valueOf(sb.toString());
    }
    public int getIndex(String gra) {
        if (gra.equals("Year")) {
            return 0;
        } else if (gra.equals("Month")) {
            return 1;
        } else if (gra.equals("Day")) {
            return 2;
        } else if (gra.equals("Hour")) {
            return 3;
        } else if (gra.equals("Minute")) {
            return 4;
        } else if (gra.equals("Second")) {
            return 5;
        }
        return -1;
    }
}
```

2. 直接用6个 `map` 分别存：虽然短但是慢

```
class LogSystem {
    map<string, map<string, vector<int>>> R;
    vector<string> G={"Year", "Month", "Day", "Hour", "Minut
e", "Second"};
public:
    LogSystem() {}
```

```
    void put(int id, string timestamp) {
        for(int i=4, j=0; i<=timestamp.size() && j<6; ++j, i+=
3){
            R[G[j]][timestamp.substr(0, i)].push_back(id);
        }
    }
    vector<int> retrieve(string s, string e, string gra) {
        vector<int> ans;
        if(R.empty()) return ans;
        int l = R[gra].begin()->first.size();
        for(auto it=R[gra].lower_bound(s.substr(0, l)); it!=R
[gra].end() && it->first<=e.substr(0, l); ++it){
            for(int k: it->second) ans.push_back(k);
        }
        return ans;
    }
};
```