

332. 重新安排行程

通过次数

提交次数

难度 中等

57

收藏

分享

切换为英文

关注

2,594

7,775

题目描述 评论(4) 题解(9) 提交记录

C++

i {} ↺ > ⌂

给定一个机票的字符串二维数组 `[from, to]`，子数组中的两个成员分别表示飞机出发和降落的机场地点，对该行程进行重新规划排序。所有这些机票都属于一个从 JFK（肯尼迪国际机场）出发的先生，所以该行程必须从 JFK 出发。

说明:

1. 如果存在多种有效的行程，你可以按字典顺序返回最小的行程组合。例如，行程 `["JFK", "LGA"]` 与 `["JFK", "LGB"]` 相比就更小，排序更靠前
2. 所有的机场都用三个大写字母表示（机场代码）。
3. 假定所有机票至少存在一种合理的行程。

示例 1:

输入: `[["MUC", "LHR"], ["JFK", "MUC"], ["SFO", "SJC"], ["LHR", "SFO"]]`
输出: `["JFK", "MUC", "LHR", "SFO", "SJC"]`

示例 2:

输入: `[["JFK", "SFO"], ["JFK", "ATL"], ["SFO", "ATL"], ["ATL", "JFK"], ["ATL", "SFO"]]`
输出: `["JFK", "ATL", "JFK", "SFO", "ATL", "SFO"]`
解释: 另一种有效的行程是 `["JFK", "SFO", "ATL", "JFK", "ATL"]`，但是它自然排序更大更靠后。

在真实的面试中遇到过这道题？

是

否

贡献者



相关企业

相关标签

```
1 class Solution {
2 public:
3     vector<string> route;
4     vector<string> findItinerary(vector<vector<string>>& tickets) {
5         /*****
6         这题本质是：求最小字典序的欧拉路径
7
8         如何判断一张有向图是否存在欧拉路径？
9         - 图是连通的
10        - 当且仅当该图的所有顶点的入度 = 出度
11        - 或者存在一个顶点的出度 = 入度+1，另一个顶点的入度 = 出度+1，且其他顶点
12
13        题目已经明确欧拉路径存在，且起点为“JFK”
14        则从起点出发，DFS，走过的边删掉，如果碰到一个顶点无法继续走下去，则说明该
15        有点贪心的意思
16
17        *****/
18        if(tickets.empty()) return {};
19
20        unordered_map<string,priority_queue<string,vector<string>,greater<
21        for(auto path:tickets){
22            string from=path[0],to=path[1];
23            graph[from].push(to);
24        }
25        dfs(graph,"JFK");
26        reverse(route.begin(),route.end());
27        return route;
28    }
29    void dfs(unordered_map<string,priority_queue<string,vector<string>,gre
30        if(graph.count(site)!=0){
31            while(graph[site].size()>0){
32                string next=graph[site].top();
33                graph[site].pop();
34                dfs(graph,next);
35            }
36        }
37        route.push_back(site);
38    }
39    };
```

