

欧拉通路/回路和一笔画问题

简介

LeetCode 每日一题刷到了一笔画问题。其核心是找到图的欧拉通路或欧拉回路。之前用DFS做的，算法也不太了解。结合题解总结一下。

欧拉通路、欧拉回路和欧拉图

欧拉通路：通过图中的每条边且只通过一次，并且经过每一顶点的通路。

欧拉回路：通过途中每条边且只通过一次，并且经过每一个顶点的回路。

无向图

欧拉通路：图连通，图中只有0个或者2个度为奇数的节点

- G仅有两个奇度节点的连通图，G的欧拉通路必以此两个节点为端点
- G无奇度节点时，必有欧拉回路

欧拉回路：图连通，图中所有节点度均为偶数。(G为欧拉图的充要条件)

具有欧拉回路的无向图为欧拉图。具有欧拉通路但不具有欧拉回路的无向图为半欧拉图。

有向图

欧拉通路：图连通；除两个端点之外其余节点入度与出度相等；1个节点的入度比出度大1，1个节点的入度比出度小1。

欧拉回路：图连通，所有节点入度等于出度。

Hierholzer 算法

Hierholzer算法给定有向图，且为欧拉图，求欧拉回路。

1. 选定起点，遍历所有边
2. DFS访问相邻顶点，将所有经过的边都删除
3. 如果当前顶点没有相邻边，顶点入栈
4. 栈中顶点倒序输出，为起始点出发的欧拉回路

如果是半欧拉图，给定起始点（入度出度相差为1）该算法可以求出欧拉通路。

LeetCode 332. 重新安排行程

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` 开始。

说明 `JFK` 肯定是起始点。直接用H算法

```
1 class Solution:
2     def findItinerary(self, tickets: List[List[str]]) ->
    List[str]:
3         ret = []
4         def dfs(cur):
5             while graph[cur]:
6                 nxt = heapq.heappop(graph[cur])
7                 dfs(nxt)
8                 ret.append(cur)
9
10        graph = collections.defaultdict(list)
11        for start, end in tickets:
12            graph[start].append(end)
13
14        for key in graph:
15            heapq.heapify(graph[key])
16        dfs('JFK')
17        return ret[::-1]
```

LeetCode 753. 破解保险箱

有一个需要密码才能打开的保险箱。密码是 n 位数，密码的每一位是 k 位序列 $0, 1, \dots, k-1$ 中的一个。

你可以随意输入密码，保险箱会自动记住最后 n 位输入，如果匹配，则能够打开保险箱。

举个例子，假设密码是 "345"，你可以输入 "012345" 来打开它，只是你输入了 6 个字符。

请返回一个能打开保险箱的最短字符串。

De Bruijn graph

```
1 class Solution:
2     def crackSafe(self, n: int, k: int) -> str:
3         seen = set()
4         highest = 10 ** (n - 1)
5         ret = []
```

```
6         def dfs(node):
7             for x in range(k):
8                 _node = node * 10 + x
9                 if _node not in seen:
10                     seen.add(_node)
11                     dfs(_node % highest)
12                     ret.append(str(x))
13
14         dfs(0)
15         # print(ret)
16         return ''.join(ret) + '0' * (n - 1)
17
```