

Assignment #B: 图为主

Updated 2223 GMT+8 Apr 29, 2025

2025 spring, Compiled by 胡新璞, 工学院

1. 题目

E07218:献给阿尔吉侬的花束

bfs, <http://cs101.openjudge.cn/practice/07218/>

思路：普通的迷宫题，套模板即可。

代码：

```
from collections import deque
def bfs(a,b,m,n):
    directions = [[1,0],[0,1],[-1,0],[0,-1]]
    q = deque([(0,a,b)])
    in_queue = {(a,b)}
    while q:
        time,x,y = q.popleft()
        if x == m and y == n:
            return time
        for i in range(len(directions)):
            nx = x + directions[i][0]
            ny = y + directions[i][1]
            if 0 <= nx < r and 0 <= ny < c and (nx,ny) not in in_queue:
                if matrix[nx][ny] != "#":
                    q.append((time + 1,nx,ny))
                    in_queue.add((nx,ny))
    return "oop!"
t = int(input())
for _ in range(t):
    r,c = map(int,input().split())
    matrix = [list(map(str,input())) for _ in range(r)]
    for i in range(r):
        for j in range(c):
            if matrix[i][j] == 'S':
                x1,y1 = i,j
            if matrix[i][j] == 'E':
                x2,y2 = i,j
    print(bfs(x1,y1,x2,y2))
```

代码运行截图 （至少包含有"Accepted"）

状态: **Accepted**

源代码

```
from collections import deque

def bfs(a,b,m,n):
    directions = [[1,0],[0,1],[-1,0],[0,-1]]
    q = deque([(0,a,b)])
    in_queue = {(a,b)}
    while q:
```

基本信息

#: 49077133
题目: 07218
提交人: 2400011037
内存: 5696kB
时间: 120ms
语言: Python3
提交时间: 2025-05-06 20:53:27

M3532.针对图的路径存在性查询 I

disjoint set, <https://leetcode.cn/problems/path-existence-queries-in-a-graph-i/>

思路：因为给的数组是非严格递增的所以只需要考虑后一个和前一个的差，有联通路径的图也是分块独立排列的，只需要判断给的数组的两个元素是不是在同一个联通块中。

代码：

class Solution:

def pathExistenceQueries(self, n: int, nums: List[int], maxDiff: int, queries: List[List[int]]) -> List[bool]:

graph_lst = [0] * n

k = 0

for i in range(1,n):

if nums[i] - nums[i-1] > maxDiff:

k += 1

graph_lst[i] += k

lst = [False] * len(queries)

for j in range(len(queries)):

if graph_lst[queries[j][0]] == graph_lst[queries[j][1]]:

lst[j] = True

return lst

代码运行截图（至少包含有"Accepted"）

通过 550 / 550 个通过的测试用例
提交于 2025.05.06 21:22

第 447 场力扣周赛
「微规划」专场，参与竞赛可获企业内推机会及精美周边，期待你的加入...

执行用时分布
123 ms | 击败 65.16%
复杂度分析

消耗内存分布
47.78 MB | 击败 72.25%

```
class Solution:
    def pathExistenceQueries(self, n: int, nums: List[int], maxDiff: int,
queries: List[List[int]]) -> List[bool]:
        graph_lst = [0] * n
        k = 0
        for i in range(1,n):
            if nums[i] - nums[i-1] > maxDiff:
                k += 1
            graph_lst[i] += k
        lst = [False] * len(queries)
        for j in range(len(queries)):
            if graph_lst[queries[j][0]] == graph_lst[queries[j][1]]:
                lst[j] = True
        return lst
```

已存储 行 14, 列 11

测试用例

Case 1 Case 2 +

n =

M22528:厚道的调分方法

binary search, <http://cs101.openjudge.cn/practice/22528/>

思路：常规的二分（有一段时间没写二分，又在判断终止那里错了几次，糟糕的）。

代码：

```
score = [float(x) for x in input().split()]
score.sort()
score_min = score[int(len(score) * 0.4)]
l, r = 1, 1000000000
mid = 0
while l < r:
    mid = (l + r) // 2
    grade = score_min * (mid / 1000000000) + 1.1 ** (score_min * (mid / 1000000000))
    if grade >= 85:
        r = mid
    else:
        l = mid + 1
print(l)
```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```
score = [float(x) for x in input().split()]
score.sort()
score_min = score[int(len(score) * 0.4)]
l, r = 1, 1000000000
mid = 0
while l < r:
    mid = (l + r) // 2
```

基本信息

#: 49077870
题目: 22528
提交人: 2400011037
内存: 18056kB
时间: 93ms
语言: Python3
提交时间: 2025-05-06 22:09:47

Msy382: 有向图判环

dfs, <https://sunnywhy.com/sfbj/10/3/382>

思路：跟着老师发的 md 文档和 AI 的指导学的拓扑排序，还不熟。

代码：

```
from collections import deque
```

```
n, m = map(int, input().split())
```

```
adj = [[] for _ in range(n)]
```

```
indegree = [0] * n
```

```
for _ in range(m):
```

```
    u, v = map(int, input().split())
```

```
    adj[u].append(v)
```

```
    indegree[v] += 1
```

```
q = deque()
```

```
for i in range(n):
```

```
    if indegree[i] == 0:
```

```
        q.append(i)
```

```
cnt = 0
```

```
while q:
```

```
    u = q.popleft()
```

```
    cnt += 1
```

```
    for v in adj[u]:
```

```
        indegree[v] -= 1
```

```
        if indegree[v] == 0:
```

```
            q.append(v)
```

```
print("Yes" if cnt != n else "No")
```

代码运行截图 （至少包含有"Accepted"）

```
14 cnt = 0
15 while q:
16     u = q.popleft()
17     cnt += 1
18     for v in adj[u]:
19         indegree[v] -= 1
20         if indegree[v] == 0:
21             q.append(v)
22
23 print("Yes" if cnt != n else "No")
```

测试输入 提交结果 历史提交

完美通过

[查看题解](#)

100% 数据通过测试 [详情](#)

运行时长: 0 ms

M05443:兔子与樱花

Dijkstra, <http://cs101.openjudge.cn/practice/05443/>

思路:

代码:

代码运行截图 (至少包含有"Accepted")

T28050: 骑士周游

dfs, <http://cs101.openjudge.cn/practice/28050/>

思路:

代码:

代码运行截图 (至少包含有"Accepted")

2. 学习总结和收获

如果发现作业题目相对简单, 有否寻找额外的练习题目, 如“数算 2025spring 每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

啊我感觉最后两题(甚至三题)对我来说都是 T 难度, 理解半天……五一主要的事情是把树给复习了一下, 主要是看讲义, 回顾往期作业的题解, 接下来打算再多做点题巩固一下, 感觉时间剩的不多了, 还是颇具挑战性的。