

Assignment #B: 图为主

Updated 2223 GMT+8 Apr 29, 2025

2025 spring, Compiled by 蔡沐轩 数学科学学院

说明:

1. 解题与记录:

对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge, Codeforces, LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora <https://typoraio.cn> 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。

2. **提交安排:** 提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。

3. **延迟提交:** 如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。

请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

E07218:献给阿尔吉侬的花束

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

思路:

BFS即可。约10min。

代码:

```
from collections import deque

directions=[(1,0),(-1,0),(0,1),(0,-1)]
for __ in range(int(input())):
    r,c=map(int,input().split())
    g=[input() for _ in range(r)]
    for i in range(r):
        for j in range(c):
            if g[i][j]=='S':x0,y0=i,j
    q=deque([(x0,y0,0)])
    inq=[[False]*c for _ in range(r)]
    inq[x0][y0]=True
    flag=False
    while q:
        x0,y0,d0=q.popleft()
        if g[x0][y0]=='E':
```

```

        print(d0)
        flag=True
        break
    for k in directions:
        x1,y1=x0+k[0],y0+k[1]
        if 0<=x1<r and 0<=y1<c and not inq[x1][y1] and g[x1][y1]!='#':
            inq[x1][y1]=True
            q.append((x1,y1,d0+1))
    if not flag:print('oop!')

```

代码运行截图 (至少包含有"Accepted")

#49032357提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

from collections import deque

directions=[(1,0),(-1,0),(0,1),(0,-1)]
for _ in range(int(input())):
    r,c=map(int,input().split())
    g=[input() for _ in range(r)]
    for i in range(r):
        for j in range(c):
            if g[i][j]=='S':x0,y0=i,j
    q=deque([(x0,y0,0)])
    inq=[[False]*c for _ in range(r)]
    inq[x0][y0]=True
    flag=False
    while q:
        x0,y0,d0=q.popleft()
        if g[x0][y0]=='E':
            print(d0)
            flag=True
            break
        for k in directions:
            x1,y1=x0+k[0],y0+k[1]
            if 0<=x1<r and 0<=y1<c and not inq[x1][y1] and g[x1][y1]!='#':
                inq[x1][y1]=True
                q.append((x1,y1,d0+1))
    if not flag:print('oop!')

```

基本信息

#: 49032357
 题目: 07218
 提交人: 24n2400010617
 内存: 4068kB
 时间: 97ms
 语言: Python3
 提交时间: 2025-04-28 22:37:04

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

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:

将 nums 排序后, 相邻的节点一定是一段连续的子数组, 于是按顺序分段, 给每段一个编号即可。约 6min。

代码:

```

class Solution:
    def pathExistenceQueries(self, n: int, nums: List[int], maxDiff: int,
queries: List[List[int]]) -> List[bool]:
        type=[-1]*n
        cur_type=0
        cur=0
        lst=sorted([(nums[i],i) for i in range(n)])
        for num,i in lst:
            if num-cur>maxDiff:

```

```

        cur_type+=1
        type[i]=cur_type
        cur=num
    ans=[]
    for u,v in queries:
        ans.append(type[u]==type[v])
    return ans

```

代码运行截图 (至少包含有"Accepted")

通过 550 / 550 个通过的测试用例

 Hyperalgebra 提交于 2025.04.27 10:42

 写题解



面向在校学生的专享特惠

完成认证享 7 折 Plus 会员，享受更多学业及职业成长帮助



⌚ 执行用时分布



176 ms | 击败 56.81% 🌿

🔮 复杂度分析

💾 消耗内存分布

56.31 MB | 击败 13.06%



代码 | Python3

```

class Solution:
    def pathExistenceQueries(self, n: int, nums: List[int], maxDiff: int, queries:
        type=[-1]*n
        cur_type=0
        cur=0
        lst=sorted([(nums[i],i) for i in range(n)])
        for num,i in lst:
            if num-cur>maxDiff:
                cur_type+=1
                type[i]=cur_type
                cur=num
        ans=[]
        for u,v in queries:
            ans.append(type[u]==type[v])
        return ans

```

⌵ 收起

M22528:厚道的调分方法

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

思路:

标准的二分。约5min。

代码:

```
N=10**9
lst=list(map(float,input().split()))
limit=(len(lst)*3-1)//5+1
l,r=0,N
def check(x):
    global N
    x=x/N
    counts=0
    for v in lst:
        if v*x+1.1**(v*x)>=85:counts+=1
    return counts>=limit

while l<r:
    mid=(l+r)//2
    if check(mid):r=mid
    else:l=mid+1
print(l)
```

代码运行截图 (至少包含有"Accepted")

#49032393提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
N=10**9
lst=list(map(float,input().split()))
limit=(len(lst)*3-1)//5+1
l,r=0,N
def check(x):
    global N
    x=x/N
    counts=0
    for v in lst:
        if v*x+1.1**(v*x)>=85:counts+=1
    return counts>=limit

while l<r:
    mid=(l+r)//2
    if check(mid):r=mid
    else:l=mid+1
print(l)
```

基本信息

#: 49032393
题目: 22528
提交人: 24n2400010617
内存: 16208kB
时间: 508ms
语言: Python3
提交时间: 2025-04-28 22:49:04

Msy382: 有向图判环

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

思路:

拓扑排序写法。约5min。

代码:

```
from collections import deque

n,m=map(int,input().split())
g=[[] for _ in range(n)]
d=[0]*n
for _ in range(m):
    u,v=map(int,input().split())
    g[u].append(v)
    d[v]+=1

counts=0
q=deque()
for i in range(n):
    if not d[i]:
        q.append(i)

while q:
    x=q.pop()
    counts+=1
    for y in g[x]:
        if d[y]:
            d[y]-=1
            if not d[y]:q.append(y)
print("No" if counts==n else "Yes")
```

代码运行截图 (至少包含有"Accepted")

```
1  from collections import deque
2
3  n,m=map(int,input().split())
4  g=[[ ] for _ in range(n)]
5  d=[0]*n
6  for _ in range(m):
7      u,v=map(int,input().split())
8      g[u].append(v)
9      d[v]+=1
10
11  counts=0
12  q=deque()
13  for i in range(n):
14      if not d[i]:
15          q.append(i)
16
17  while q:
18      x=q.pop()
19      counts+=1
20      for y in g[x]:
21          if d[y]:
```

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:

Dijkstra, 同时记录最短路线中每个节点的父节点, 即可输出最短路线。

代码:

```
from heapq import *

n= int(input())
spot=[input() for _ in range(n)]
index={spot[i]:i for i in range(len(spot))}
```

```

g = [[] for _ in range(n)]
q = []

m=int(input())
for _ in range(m):
    u, v, c = input().split()
    u,v,c=index[u],index[v],int(c)
    g[u].append((c, v))
    g[v].append((c, u))

t=int(input())
for _ in range(t):
    a,b=input().split()
    a,b=index[a],index[b]
    d = [float('inf') for _ in range(n)]
    d1=[float('inf') for _ in range(n)]
    check = [0 for _ in range(n)]
    father = [0 for _ in range(n)]
    heappush(q, (0, b))
    d[b] = 0
    while q:
        x = heappop(q)[1]
        if not check[x]:
            check[x] = 1
            for r in g[x]:
                if r[0] + d[x] < d[r[1]] and not check[r[1]]:
                    d[r[1]] = r[0] + d[x]
                    d1[r[1]]=r[0]
                    heappush(q, (d[r[1]], r[1]))
                    father[r[1]] = x
    while a != b:
        print(f"{spot[a]}->({d1[a]})->",end='')
        a = father[a]
    print(spot[a])

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
from heapq import *

n= int(input())
spot=[input() for _ in range(n)]
index={spot[i]:i for i in range(len(spot))}
g = [[] for _ in range(n)]
q = []

m=int(input())
for _ in range(m):
    u, v, c = input().split()
    u,v,c=index[u],index[v],int(c)
    g[u].append((c, v))
    g[v].append((c, u))

t=int(input())
for _ in range(t):
    a,b=input().split()
    a,b=index[a],index[b]
    d = [float('inf') for _ in range(n)]
    d1=[float('inf') for _ in range(n)]
    check = [0 for _ in range(n)]
    father = [0 for _ in range(n)]
    heappush(q, (0, b))
    d[b] = 0
    while q:
        x = heappop(q)[1]
        if not check[x]:
            check[x] = 1
            for r in g[x]:
                if r[0] + d[x] < d[r[1]] and not check[r[1]]:
                    d[r[1]] = r[0] + d[x]
                    d1[r[1]]=r[0]
                    heappush(q, (d[r[1]], r[1]))
                    father[r[1]] = x
    while a != b:
        print(f"{spot[a]}->({d1[a]})->",end='')
        a = father[a]
    print(spot[a])
```

基本信息

#: 49044177
题目: 05443
提交人: 24n2400010617
内存: 3728kB
时间: 24ms
语言: Python3
提交时间: 2025-05-01 11:02:09

T28050: 骑士周游

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

思路:

DFS, 优先走下一步走法最少的格子即可。

代码:

```
directions=[(2,1),(1,2),(-2,1),(-1,2),(2,-1),(1,-2),(-2,-1),(-1,-2)]
n=int(input())
p,q=map(int,input().split())
visited=[[1]*(n+4) for _ in range(2)]+[[1,1]+[0]*n+[1,1] for _ in range(n)]+[[1]*(n+4) for _ in range(2)]
l,b=1,0

def dfs(i:int,j:int):
    global l,n,b
    if b:return
    if l==n*n:
        b=1
        return
    visited[i][j] = 1
    mn=9
    for k1 in directions:
        x, y = i + k1[0], j + k1[1]
```



```

count=0
if not visited[x][y]:
    for k2 in directions:
        if not visited[x + k2[0]][y + k2[1]]:
            count+=1
            if count<mn:x0,y0,mn=x,y,count
if mn==9:return
l+=1
dfs(x0,y0)
l-=1
visited[i][j] = 0
return

dfs(p+2,q+2)
if b:print("success")
else:print("fail")

```

代码运行截图 (至少包含有"Accepted")

#49044193提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

directions=[(2,1),(1,2),(-2,1),(-1,2),(2,-1),(1,-2),(-2,-1),(-1,-2)]
n=int(input())
p,q=map(int,input().split())
visited=[[1]*(n+4) for _ in range(2)]+[[1,1]+[0]*n+[1,1] for _ in range(
1,b=1,0

def dfs(i:int,j:int):
    global l,n,b
    if b:return
    if l==n*n:
        b=1
        return
    visited[i][j] = 1
    mn=9
    for k1 in directions:
        x, y = i + k1[0], j + k1[1]
        count=0
        if not visited[x][y]:
            for k2 in directions:
                if not visited[x + k2[0]][y + k2[1]]:
                    count+=1
                    if count<mn:x0,y0,mn=x,y,count
            if mn==9:return
        l+=1
        dfs(x0,y0)
        l-=1
    visited[i][j] = 0
    return

dfs(p+2,q+2)
if b:print("success")
else:print("fail")

```

基本信息

#: 49044193
 题目: 28050
 提交人: 24n2400010617
 内存: 3800kB
 时间: 22ms
 语言: Python3
 提交时间: 2025-05-01 11:08:07

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

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。

作业题目还是比较轻松的，就复习了一下寒假学的图算法。依旧保持完成每日选做的新题和LeetCode每日一题。

上周LeetCode双周赛和周赛惨遭卡常，在双周赛P4和周赛P3上耽误了大量时间调整。这种题目拿C++写思路完全一致的代码就能顺利通过，感觉这方面python的时间劣势还是比较明显的。以后估计还得多加注意缩减常数。