

Assignment #B: 图为主

Updated 2223 GMT+8 Apr 29, 2025

2025 spring, Compiled by 郑涵予 物理学院

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:

经典模板题, 不过代码还是挺长的(用时约5min)

代码:

```
from collections import deque
directions=[(-1,0),(1,0),(0,-1),(0,1)]
def bfs(x1,y1,x2,y2):
    q=deque()
    q.append((x1,y1,0))
    while q:
        x,y,s=q.popleft()
        if x==x2 and y==y2:
            return s
        for dx,dy in directions:
            nx,ny=x+dx,y+dy
            if 0<=nx<m and 0<=ny<n and not visited[nx][ny] and a[nx][ny]!='#':
                visited[nx][ny]=True
                q.append((nx,ny,s+1))
    return 'oop!'
t=int(input())
```

```

for _ in range(t):
    m,n=map(int,input().split())
    a=[]
    visited=[[False]*n for _ in range(m)]
    for _ in range(m):
        a.append(input())
    x1=y1=x2=y2=-1
    for i in range(m):
        for j in range(n):
            if a[i][j]=='S':
                x1,y1=i,j
            if a[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
directions=[(-1,0),(1,0),(0,-1),(0,1)]
def bfs(x1,y1,x2,y2):
    q=deque()
    q.append((x1,y1,0))
    while q:
        x,y,s=q.popleft()
        if x==x2 and y==y2:
            return s
        for dx,dy in directions:
            nx,ny=x+dx,y+dy
            if 0<=nx<m and 0<=ny<n and not visited[nx][ny] and a[nx][ny]!='S':
                visited[nx][ny]=True
                q.append((nx,ny,s+1))
    return 'oop!'
t=int(input())
for _ in range(t):
    m,n=map(int,input().split())
    a=[]
    visited=[[False]*n for _ in range(m)]
    for _ in range(m):
        a.append(input())
    x1=y1=x2=y2=-1
    for i in range(m):
        for j in range(n):
            if a[i][j]=='S':
                x1,y1=i,j
            if a[i][j]=='E':
                x2,y2=i,j

```

基本信息

#: 49032808
 题目: 07218
 提交人: 24n2400011325
 内存: 4380kB
 时间: 92ms
 语言: Python3
 提交时间: 2025-04-29 08:06:42

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:

这里数组已经被排好序了, 所以只要直接判断相邻的两个数之差会不会大于maxDiff就行, 如果大于就把后一个数归入下一组.接下来只要判断查询的数是不是在同一个组里就行.(用时约8min)

代码:

```

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

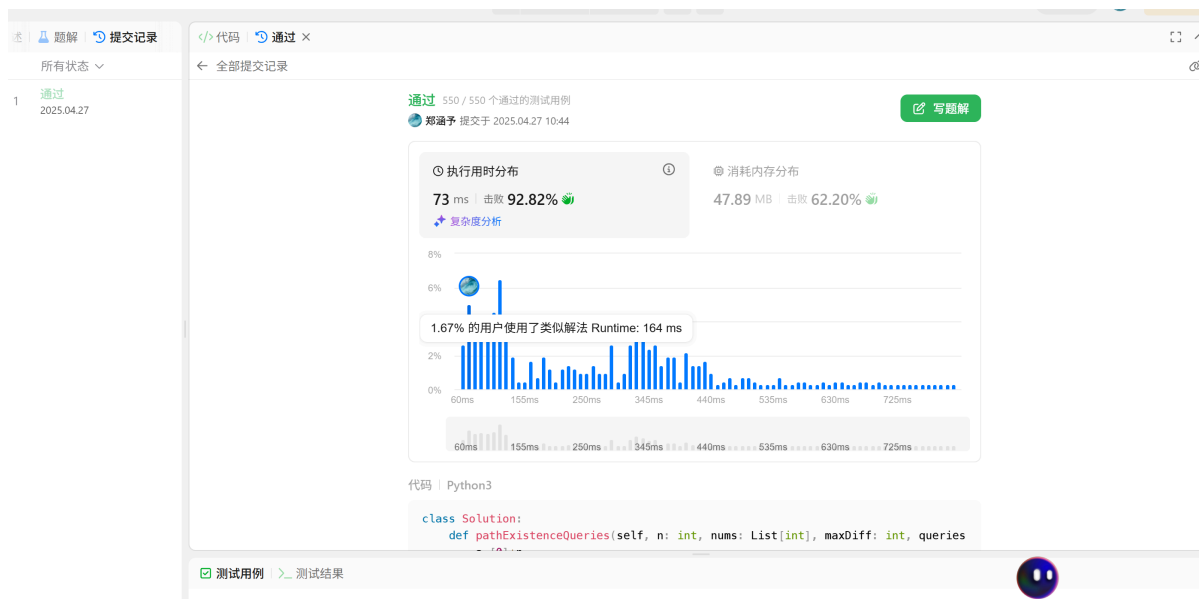
```

```

a=[0]*n
pos=0
for i in range(n-1):
    if nums[i+1]-nums[i]<=maxDiff:
        a[i]=a[i+1]=pos
    else:
        a[i]=pos
        pos+=1
        a[i+1]=pos
m=len(queries)
ans=[False]*m
for i in range(m):
    u,v=queries[i][0],queries[i][1]
    if a[u]==a[v]:
        ans[i]=True
    else:
        ans[i]=False
return ans

```

代码运行截图 (至少包含有"Accepted")



M22528:厚道的调分方法

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

思路:

因为这是保序调分, 所以首先找出恰好在60%位置上的人, 接下来其实就是二分法求解方程.(用时约10min)

代码:

```

a=list(map(float,input().split()))
p=10**9
n=len(a)
a.sort()

```

```

pos=int(n*0.4)
temp=a[pos]
def f(x,t):
    return t*x+1.1**(t*x)
left,right=1,p
while left<right:
    mid=(left+right-1)//2
    if f(temp,mid/p)>=85:
        right=mid
    else:
        left=mid+1
print(left)

```

代码运行截图 (至少包含有"Accepted")

#49032813提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

a=list(map(float,input().split()))
p=10**9
n=len(a)
a.sort()
pos=int(n*0.4)
temp=a[pos]
def f(x,t):
    return t*x+1.1**(t*x)
left,right=1,p
while left<right:
    mid=(left+right-1)//2
    if f(temp,mid/p)>=85:
        right=mid
    else:
        left=mid+1
print(left)

```

基本信息

#: 49032813
 题目: 22528
 提交人: 24n2400011325
 内存: 16248kB
 时间: 88ms
 语言: Python3
 提交时间: 2025-04-29 08:21:09

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

Msy382: 有向图判环

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

思路:

可以用染色法判断, 也可以拓扑排序 (用时约5min)

代码:

```

n,m=map(int,input().split())
a=[[ ] for _ in range(n)]
for _ in range(m):
    u,v=map(int,input().split())
    a[u].append(v)
delta=False
vis=[0]*n

def dfs(pos):
    for x in a[pos]:
        if not vis[x]:
            vis[x]=1

```

```

        dfs(x)
    if vis[x]==1:
        global delta
        delta=True
    vis[pos]=2

for i in range(n):
    if not vis[i]:
        vis[i]=1
        dfs(i)
print('Yes' if delta else 'No')

```

代码运行截图 (至少包含有"Accepted")

代码书写

Python

```

4      u,v=map(int,input().split())
5      a[u].append(v)
6      delta=False
7      vis=[0]*n
8
9      def dfs(pos):
10         for x in a[pos]:
11             if not vis[x]:
12                 vis[x]=1
13                 dfs(x)
14             if vis[x]==1:
15                 global delta
16                 delta=True
17         vis[pos]=2
18
19     for i in range(n):
20         if not vis[i]:
21             vis[i]=1
22             dfs(i)
23     print('Yes' if delta 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

收起面板

运行

提交

M05443:兔子与樱花

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

思路:

这也是经典的模板题了，直接做就行（用时约10min）

代码:

```
import heapq
n=int(input())
s=['']*n
mp={}
for i in range(n):
    s[i]=input()
    mp[s[i]]=i
a=[]
for _ in range(n):
    a.append([])
m=int(input())
for _ in range(m):
    u,v,w=input().split()
    a[mp[u]].append((mp[v],int(w)))
    a[mp[v]].append((mp[u],int(w)))

def Dijkstra(x1,x2):
    q=[]
    dist=[float('inf')]*n
    dist[x1]=0
    pre=[(-1,-1)]*n
    heapq.heappush(q,(0,x1))
    while q:
        d,p=heapq.heappop(q)
        if p==x2:
            break
        for x,c in a[p]:
            if dist[x]>dist[p]+c:
                dist[x]=dist[p]+c
                pre[x]=(p,c)
                heapq.heappush(q,(dist[x],x))
    path=[]
    while x2!=-1:
        path.append(s[x2])
        path.append('('+str(pre[x2][1])+')')
        x2=pre[x2][0]
    return '->'.join(path[::-1][1::])

t=int(input())
for _ in range(t):
    st,ed=input().split()
    print(Dijkstra(mp[st],mp[ed]))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
import heapq
n=int(input())
s=['']*n
mp={}
for i in range(n):
    s[i]=input()
    mp[s[i]]=i
a=[[] for _ in range(n)]
m=int(input())
for _ in range(m):
    u,v,w=input().split()
    a[mp[u]].append((mp[v],int(w)))
    a[mp[v]].append((mp[u],int(w)))

def Dijkstra(x1,x2):
    q=[]
    dist=[float('inf')]*n
    dist[x1]=0
    pre=[-1,-1]*n
    heapq.heappush(q,(0,x1))
    while q:
        d,p=heapq.heappop(q)
        if p==x2:
```

基本信息

#: 49026963
题目: 05443
提交人: 24n2400011325
内存: 3648kB
时间: 22ms
语言: Python3
提交时间: 2025-04-27 22:45:13

T28050: 骑士周游

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

思路:

用到了启发式搜索, 这个如果没看书真的不是自己能想出来的. 看完后用上这个算法之后做起来就很快. (用时约20min)

代码:

```
n=int(input())
x0,y0=map(int,input().split())
a=[[False]*n for _ in range(n)]
directions=[(2,1),(2,-1),(1,2),(1,-2),(-2,1),(-2,-1),(-1,2),(-1,-2)]
delta=False
a[x0][y0]=True
def calculate(x,y):
    num=0
    for dx,dy in directions:
        nx,ny=x+dx,y+dy
        if 0<=nx<n and 0<=ny<n and a[nx][ny]==False:
            num+=1
    return num
def dfs(x,y,step):
    if step==n*n:
        global delta
        delta=True
        if delta:return
    b=[]
    for dx,dy in directions:
        nx,ny=x+dx,y+dy
        if 0<=nx<n and 0<=ny<n and a[nx][ny]==False:
            b.append((calculate(nx,ny),nx,ny))
    b.sort()
    for x in b:
        a[x[1]][x[2]]=True
```

```
        dfs(x[1],x[2],step+1)
        a[x[1]][x[2]]=False
dfs(x0,y0,1)
print("success" if delta else "fail")
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n=int(input())
x0,y0=map(int,input().split())
a=[[False]*n for _ in range(n)]
directions=[(2,1),(2,-1),(1,2),(1,-2),(-2,1),(-2,-1),(-1,2),(-1,-2)]
delta=False
a[x0][y0]=True
def calculate(x,y):
    num=0
    for dx,dy in directions:
        nx,ny=x+dx,y+dy
        if 0<=nx<n and 0<=ny<n and a[nx][ny]==False:
            num+=1
    return num
def dfs(x,y,step):
    if step==n*n:
        global delta
        delta=True
        if delta:return
    b=[]
    for dx,dy in directions:
        nx,ny=x+dx,y+dy
        if 0<=nx<n and 0<=ny<n and a[nx][ny]==False:
            b.append((calculate(nx,ny),nx,ny))
    b.sort()
    for x in b:
        a[x[1]][x[2]]=True
        dfs(x[1],x[2],step+1)
        a[x[1]][x[2]]=False
dfs(x0,y0,1)
print("success" if delta else "fail")
```

基本信息

#: 49038624
题目: 28050
提交人: 24n2400011325
内存: 3908kB
时间: 25ms
语言: Python3
提交时间: 2025-04-29 22:49:18

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。

这周力扣周赛似乎是企业赞助场所以比较难, 不过感觉第二题似乎给了第四题一些提示, 还是能顺着思路做出第四题的, 反倒是第三题的dp没写出来(不过数算应该也不考了吧)。但是考场上要切难题感觉心理压力还是不小, 希望机考的T没有力扣上那么难.cheat sheet基本整理完了, 五一后月考试着用下。