Assignment #B: 图为主

Updated 2223 GMT+8 Apr 29, 2025

2025 spring, Complied by 同学的姓名、院系

> **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/

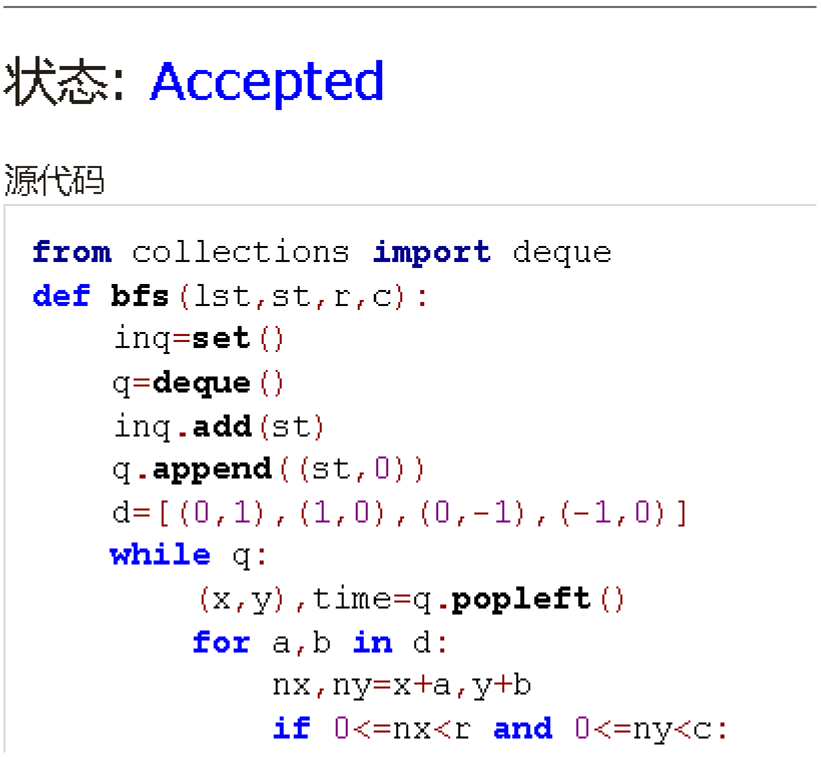
思路：

代码：

```python
from collections import deque
def bfs(lst,st,r,c):
    inq=set()
    q=deque()
    inq.add(st)
    q.append((st,0))
    d=[(0,1),(1,0),(0,-1),(-1,0)]
    while q:
        (x,y),time=q.popleft()
        for a,b in d:
            nx,ny=x+a,y+b
            if 0<=nx<r and 0<=ny<c:
                if lst[nx][ny]=='E':
                    return time+1
                elif lst[nx][ny]=='.' and (nx,ny) not in inq:
                    inq.add((nx,ny))
                    q.append(((nx,ny),time+1))
    return 'oop!'

t=int(input())
for _ in range(t):
    r,c=[int(x) for x in input().split()]
    lst=[]
    for __ in range(r):
        s=input()
        lst.append(s)
        if 'S' in s:
            st=(__,s.index('S'))
    print(bfs(lst,st,r,c))
```

代码运行截图 （至少包含有"Accepted"）


状态: Accepted

源代码

```
from collections import deque
def bfs(lst,st,r,c):
    inq=set()
    q=deque()
    inq.add(st)
    q.append((st,0))
    d=[(0,1),(1,0),(0,-1),(-1,0)]
    while q:
        (x,y),time=q.popleft()
        for a,b in d:
            nx,ny=x+a,y+b
            if 0<=nx<r and 0<=ny<c:
```

M3532.针对图的路径存在性查询I

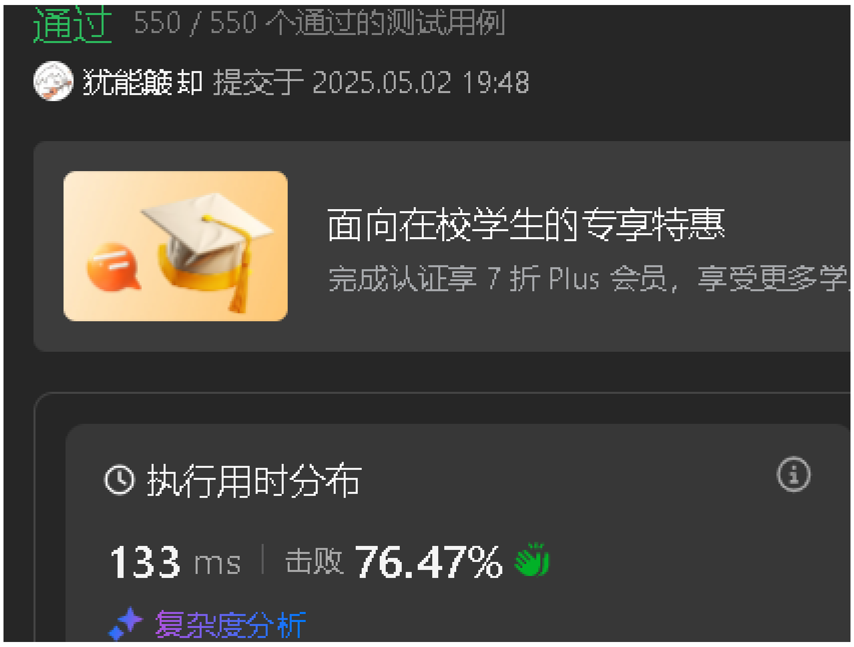
disjoint set, https://leetcode.cn/problems/path-existence-queries-in-a-graph-i/

思路：

代码：

```
class Solution(object):
    def pathExistenceQueries(self, n, nums, maxDiff, queries):
        """
        :type n: int
        :type nums: List[int]
        :type maxDiff: int
        :type queries: List[List[int]]
        :rtype: 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
        ans=[False]*len(queries)
        for i in range(len(queries)):
            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"）



通过 550 / 550 个通过的测试用例

犹能簸却 提交于 2025.05.02 19:48

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

执行用时分布

133 ms | 击败 76.47%

复杂度分析

M22528:厚道的调分方法

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

思路：

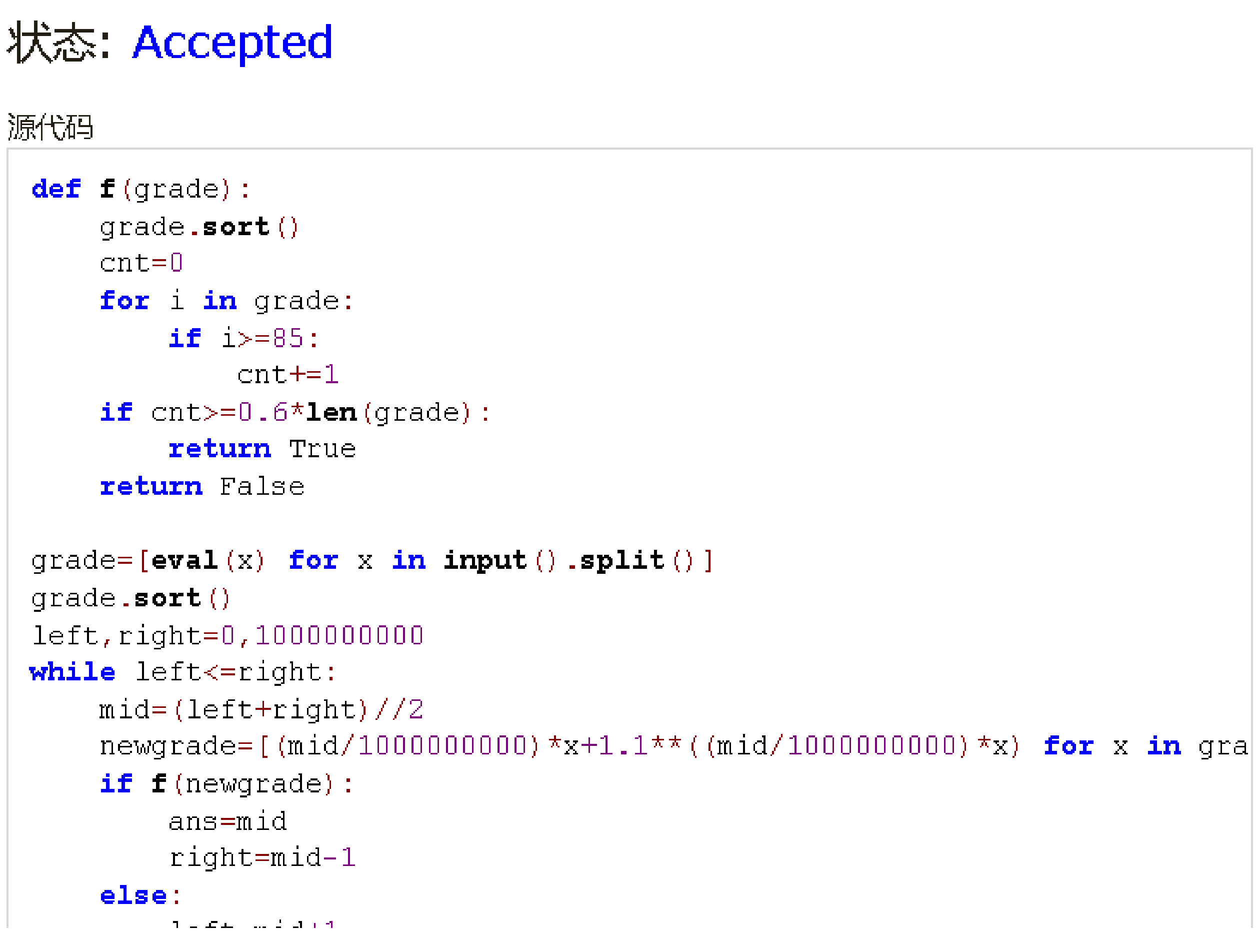
代码：

```python
def f(grade):
    grade.sort()
    cnt=0
    for i in grade:
        if i>=85:
            cnt+=1
    if cnt>=0.6*len(grade):
        return True
    return False

grade=[eval(x) for x in input().split()]
grade.sort()
left,right=0,1000000000
while left<=right:
    mid=(left+right)//2
    newgrade=[(mid/1000000000)*x+1.1**((mid/1000000000)*x) for x in grade]
    if f(newgrade):
        ans=mid
        right=mid-1
    else:
        left=mid+1

print(ans)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
def f(grade):
    grade.sort()
    cnt=0
    for i in grade:
        if i>=85:
            cnt+=1
    if cnt>=0.6*len(grade):
        return True
    return False

grade=[eval(x) for x in input().split()]
grade.sort()
left,right=0,1000000000
while left<=right:
    mid=(left+right)//2
    newgrade=[(mid/1000000000)*x+1.1**((mid/1000000000)*x) for x in gra
    if f(newgrade):
        ans=mid
        right=mid-1
    else:
        left=mid+1
```

Msy382: 有向图判环

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

思路:

代码:

```python
from collections import defaultdict
import sys
sys.setrecursionlimit(1<<30)
def f(graph):
    def dfs(key,st,step):
        if key==st and step:
            return True
        lst=graph[key]
        for i in lst:
            if i not in visited:
                visited.add(i)
                if dfs(i,st,True):
                    return True
        return False
    keys=list(graph.keys())
    for key in keys:
        visited=set()
        if dfs(key,key,False):
            return 'Yes'
    return 'No'

n,m=[int(x) for x in input().split()]
graph=defaultdict(list)
for _ in range(m):
    u,v=[int(x) for x in input().split()]
    graph[u].append(v)
print(f(graph))
```

代码运行截图 （至少包含有"Accepted"）

M05443:兔子与樱花

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

思路：

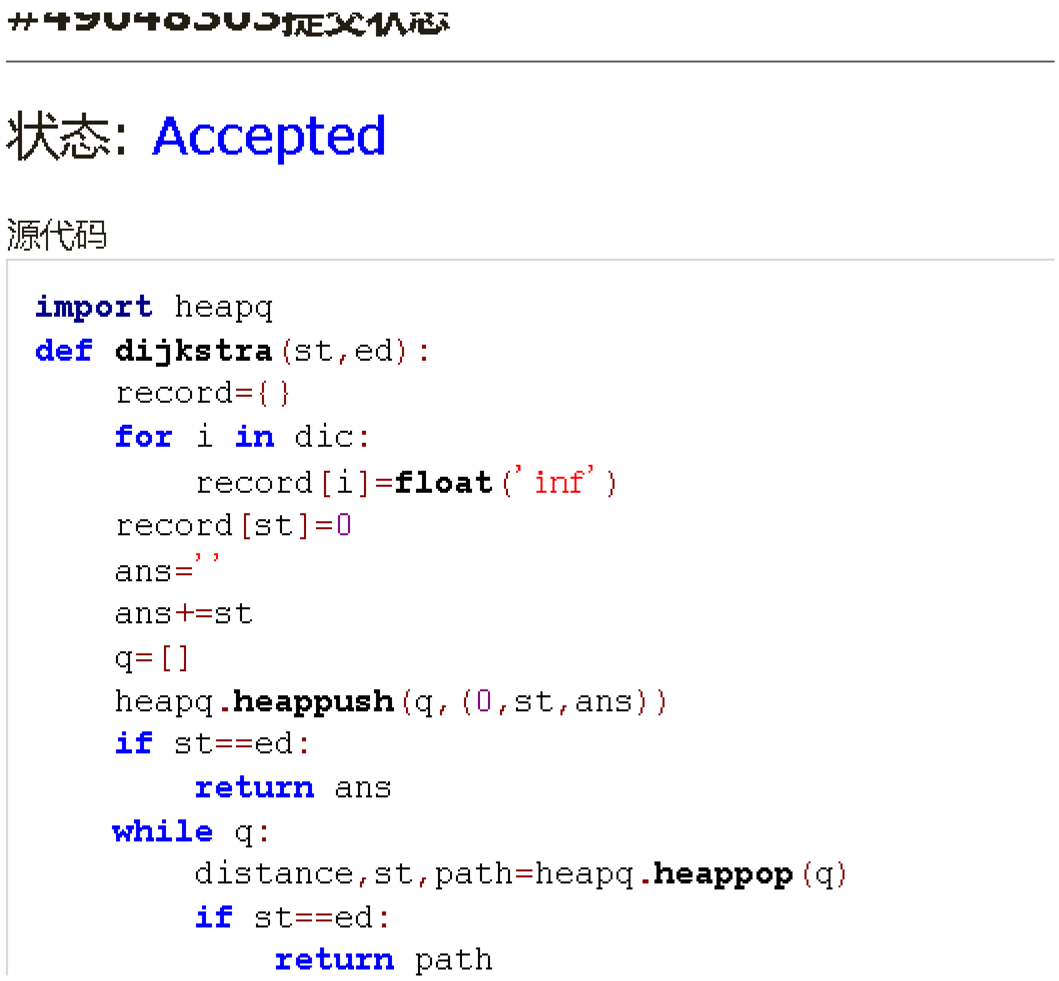
代码：

```
import heapq
def dijkstra(st,ed):
    record={}
    for i in dic:
        record[i]=float('inf')
    record[st]=0
    ans=''
    ans+=st
    q=[]
    heapq.heappush(q,(0,st,ans))
    if st==ed:
        return ans
    while q:
        distance,st,path=heapq.heappop(q)
        if st==ed:
            return path
        lst=dic[st]
        for next,d in lst:
            if distance<=record[st]:
                record[st]=distance
                newpath=path+'->('+str(d)+')->'+next
                heapq.heappush(q,(distance+d,next,newpath))


p=int(input())
dic={}
for _ in range(p):
    dic[input()]=[]
```

```python
q=int(input())
for _ in range(q):
    a,b,c=[x for x in input().split()]
    c=int(c)
    dic[a].append((b,c))
    dic[b].append((a,c))
r=int(input())
for _ in range(r):
    st,ed=input().split()
    print(dijkstra(st,ed))
```

代码运行截图 （至少包含有"Accepted"）



#49048505提交状态

状态: Accepted

源代码

```python
import heapq
def dijkstra(st,ed):
    record={}
    for i in dic:
        record[i]=float('inf')
    record[st]=0
    ans=''
    ans+=st
    q=[]
    heapq.heappush(q,(0,st,ans))
    if st==ed:
        return ans
    while q:
        distance,st,path=heapq.heappop(q)
        if st==ed:
            return path
```

T28050: 骑士周游

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

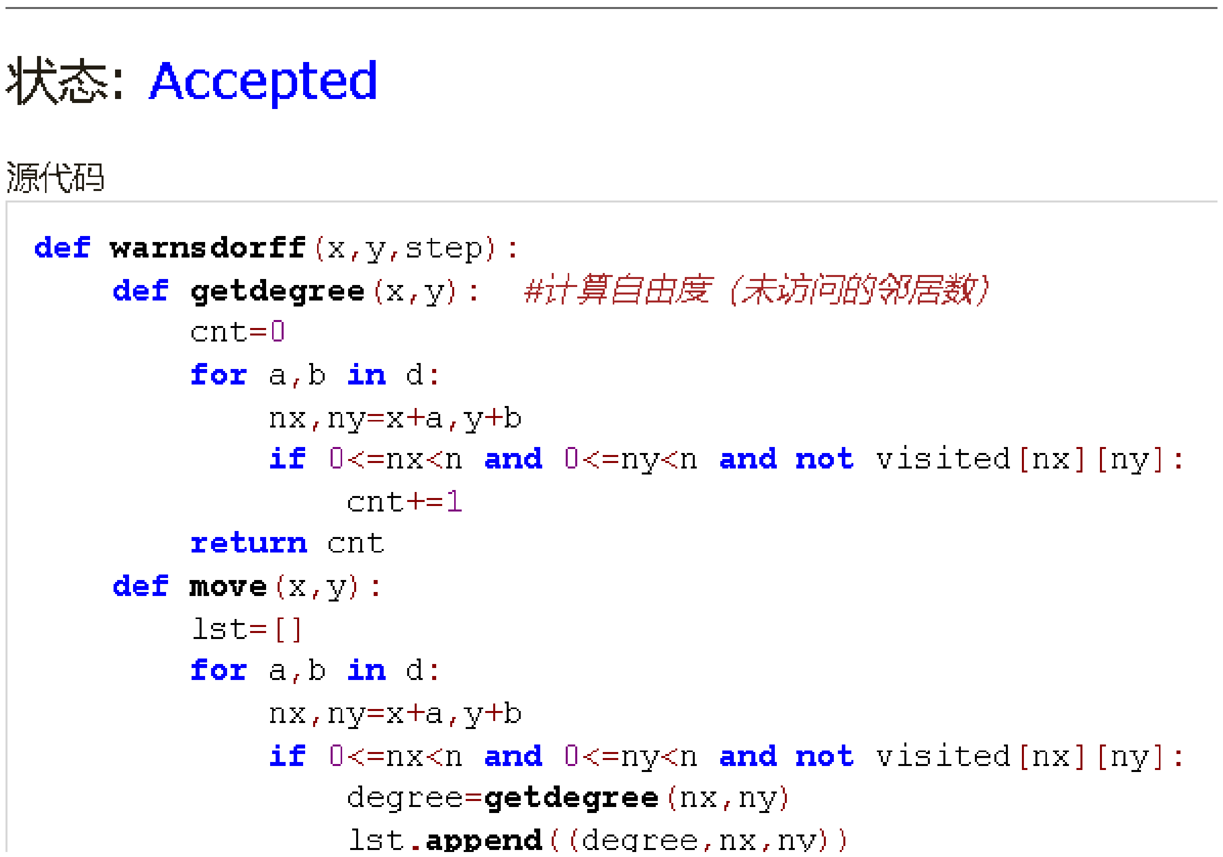
思路：学习了Warnsdorff规则，不然总是超时......

代码：

```python
def warnsdorff(x,y,step):
    def getdegree(x,y):  #计算自由度（未访问的邻居数）
        cnt=0
        for a,b in d:
            nx,ny=x+a,y+b
            if 0<=nx<n and 0<=ny<n and not visited[nx][ny]:
                cnt+=1
        return cnt
    def move(x,y):
        lst=[]
        for a,b in d:
            nx,ny=x+a,y+b
            if 0<=nx<n and 0<=ny<n and not visited[nx][ny]:
                degree=getdegree(nx,ny)
                lst.append((degree,nx,ny))
        if not lst:
            return None
        lst.sort()
        return lst[0][1],lst[0][2]   #返回自由度最小的两个点
    while step<n*n:
        nextmove=move(x,y)
        if not nextmove:
            return 'fail'
        x,y=nextmove
        visited[x][y]=True
        step+=1
    return 'success'

n=int(input())
x,y=[int(x) for x in input().split()]
visited=[[False]*n for _ in range(n)]
visited[x][y]=True
d=[(1,2),(1,-2),(-1,2),(-1,-2),(2,1),(2,-1),(-2,1),(-2,-1)]
print(warnsdorff(x,y,1))
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```python
def warnsdorff(x,y,step):
    def getdegree(x,y):   #计算自由度（未访问的邻居数）
        cnt=0
        for a,b in d:
            nx,ny=x+a,y+b
            if 0<=nx<n and 0<=ny<n and not visited[nx][ny]:
                cnt+=1
        return cnt
    def move(x,y):
        lst=[]
        for a,b in d:
            nx,ny=x+a,y+b
            if 0<=nx<n and 0<=ny<n and not visited[nx][ny]:
                degree=getdegree(nx,ny)
                lst.append((degree,nx,ny))
```

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。

终于考完期中了，我将在五一恶补数算，五一快乐！！