Assignment #A: 图论：算法，树算及栈

Updated 2018 GMT+8 Apr 21, 2024

2024 spring, Complied by 田济维 物理学院

说明：

1）请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora https://typoraio.cn ，或者用word）。AC 或者没有AC，都请标上每个题目大致花费时间。

2）提交时候先提交pdf文件，再把md或者doc文件上传到右侧"作业评论"。Canvas需要有同学清晰头像、提交文件有pdf、"作业评论"区有上传的md或者doc附件。

3）如果不能在截止前提交作业，请写明原因。

编程环境

（python pycharm）

操作系统：macOS Ventura 13.4.1 (c)

Python编程环境：Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境：Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

1. 题目

20743: 整人的提词本

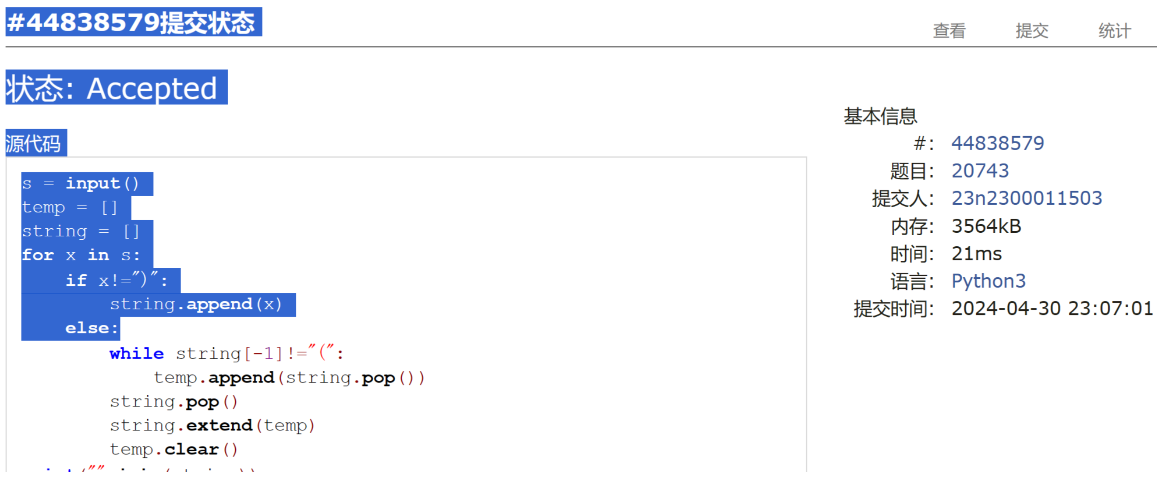
http://cs101.openjudge.cn/practice/20743/

思路：

代码

```python
#
s = input()
temp = []
string = []
for x in s:
    if x!=")":
        string.append(x)
    else:
        while string[-1]!="(":
            temp.append(string.pop())
        string.pop()
        string.extend(temp)
        temp.clear()
print("".join(string))
```

代码运行截图 （至少包含有"Accepted"）



#44838579提交状态

查看　提交　统计

状态: Accepted

源代码

```python
s = input()
temp = []
string = []
for x in s:
    if x!=")":
        string.append(x)
    else:
        while string[-1]!="(":
            temp.append(string.pop())
        string.pop()
        string.extend(temp)
        temp.clear()
```

基本信息

#:	44838579
题目:	20743
提交人:	23n2300011503
内存:	3564kB
时间:	21ms
语言:	Python3
提交时间:	2024-04-30 23:07:01

02255: 重建二叉树

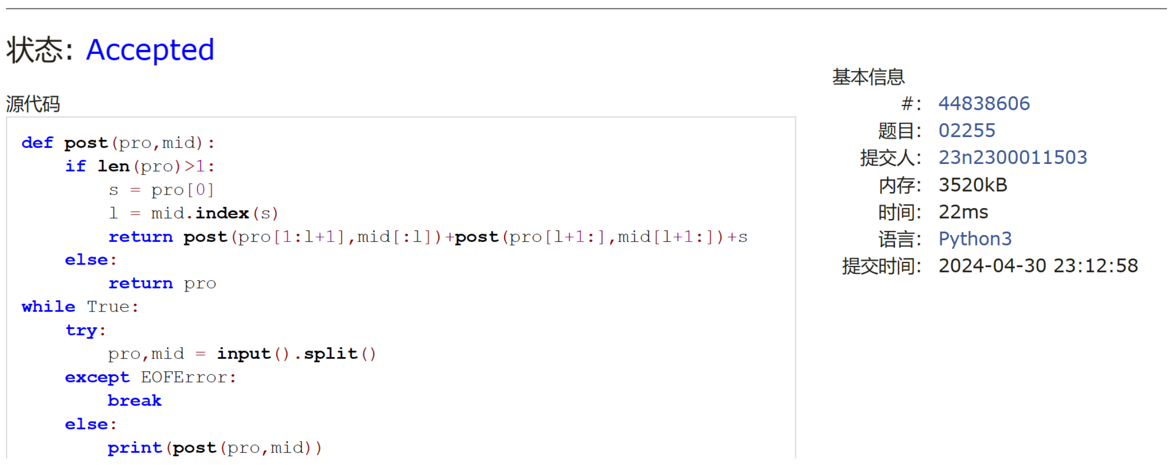
http://cs101.openjudge.cn/practice/02255/

思路:

代码

```python
#
def post(pro,mid):
    if len(pro)>1:
        s = pro[0]
        l = mid.index(s)
        return post(pro[1:l+1],mid[:l])+post(pro[l+1:],mid[l+1:])+s
    else:
        return pro
while True:
    try:
        pro,mid = input().split()
    except EOFError:
        break
    else:
        print(post(pro,mid))
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```python
def post(pro,mid):
    if len(pro)>1:
        s = pro[0]
        l = mid.index(s)
        return post(pro[1:l+1],mid[:l])+post(pro[l+1:],mid[l+1:])+s
    else:
        return pro
while True:
    try:
        pro,mid = input().split()
    except EOFError:
        break
    else:
        print(post(pro,mid))
```

基本信息

#: 44838606
题目: 02255
提交人: 23n2300011503
内存: 3520kB
时间: 22ms
语言: Python3
提交时间: 2024-04-30 23:12:58

01426: Find The Multiple

http://cs101.openjudge.cn/practice/01426/

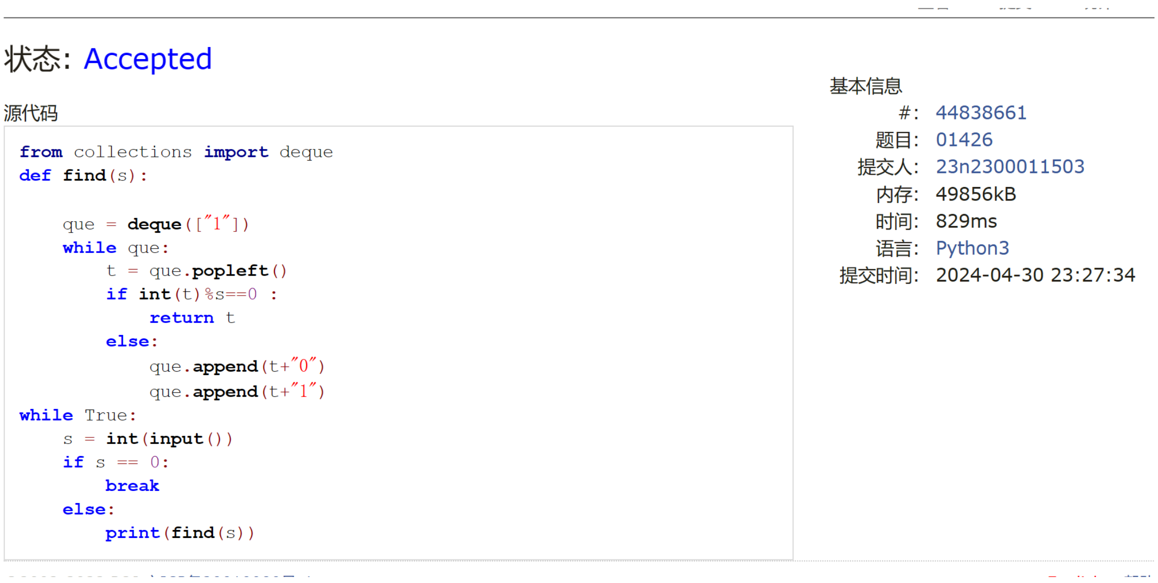
要求用bfs实现

思路:

代码

```python
#
from collections import deque
def find(s):

    que = deque(["1"])
    while que:
        t = que.popleft()
        if int(t)%s==0 :
            return t
        else:
            que.append(t+"0")
            que.append(t+"1")
while True:
    s = int(input())
    if s == 0:
        break
    else:
        print(find(s))

```

代码运行截图 (AC代码截图，至少包含有"Accepted")


状态: Accepted

源代码

```python
from collections import deque
def find(s):

    que = deque(["1"])
    while que:
        t = que.popleft()
        if int(t)%s==0 :
            return t
        else:
            que.append(t+"0")
            que.append(t+"1")
while True:
    s = int(input())
    if s == 0:
        break
    else:
        print(find(s))
```

基本信息

#:	44838661
题目:	01426
提交人:	23n2300011503
内存:	49856kB
时间:	829ms
语言:	Python3
提交时间:	2024-04-30 23:27:34

04115: 鸣人和佐助

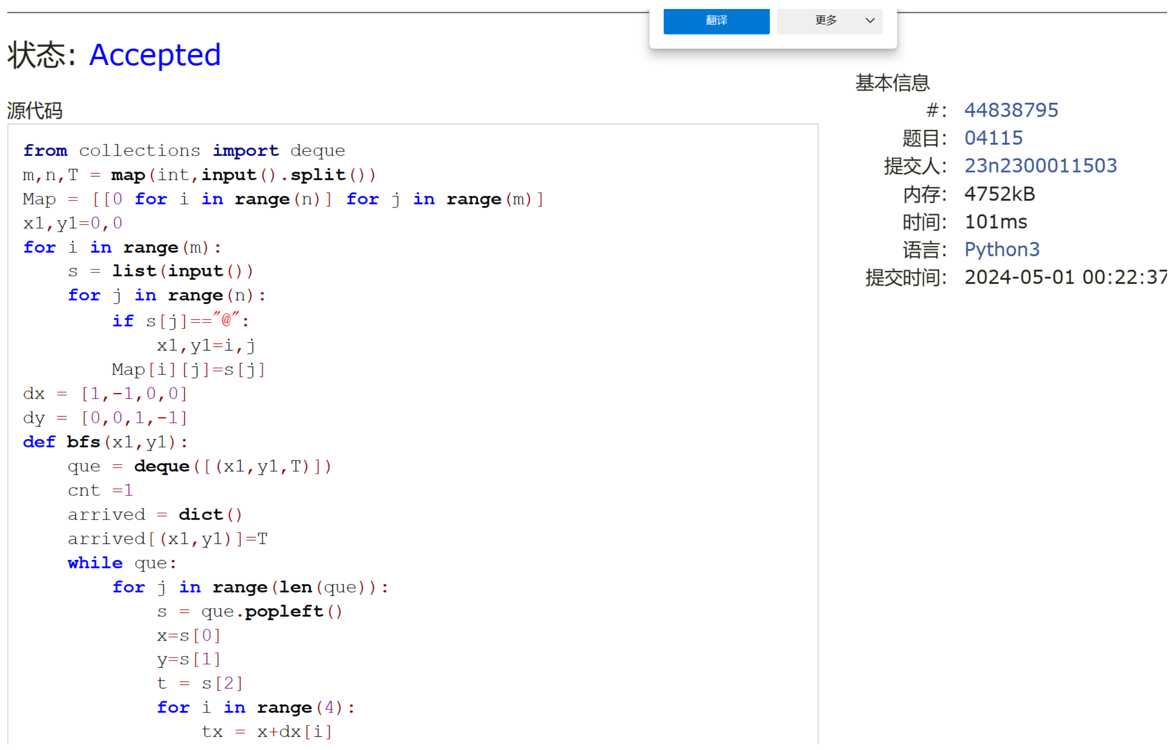
bfs, http://cs101.openjudge.cn/practice/04115/

思路:

代码

```python
#
from collections import deque
m,n,T = map(int,input().split())
Map = [[0 for i in range(n)] for j in range(m)]
x1,y1=0,0
for i in range(m):
    s = list(input())
    for j in range(n):
        if s[j]=="@":
            x1,y1=i,j
        Map[i][j]=s[j]
dx = [1,-1,0,0]
dy = [0,0,1,-1]
def bfs(x1,y1):
    que = deque([(x1,y1,T)])
    cnt =1
    arrived = dict()
    arrived[(x1,y1)]=T
    while que:
        for j in range(len(que)):
            s = que.popleft()
            x=s[0]
            y=s[1]
            t = s[2]
            for i in range(4):
                tx = x+dx[i]
                ty = y+dy[i]
                if 0<=tx<m and 0<=ty<n and ((tx,ty) not in arrived or arrived[(tx,ty)]<t):
                    if Map[tx][ty]=="*":
                        que.append((tx,ty,t))
                        arrived[(tx,ty)]=t
                    elif Map[tx][ty]=="#":
                        if t>0:
                            que.append((tx,ty,t-1))
                            arrived[(tx,ty)]=t-1
                    elif Map[tx][ty]=="+":
                        return cnt
        cnt+=1
    return -1
print(bfs(x1,y1))
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```
from collections import deque
m,n,T = map(int,input().split())
Map = [[0 for i in range(n)] for j in range(m)]
x1,y1=0,0
for i in range(m):
    s = list(input())
    for j in range(n):
        if s[j]=="@":
            x1,y1=i,j
        Map[i][j]=s[j]
dx = [1,-1,0,0]
dy = [0,0,1,-1]
def bfs(x1,y1):
    que = deque([(x1,y1,T)])
    cnt =1
    arrived = dict()
    arrived[(x1,y1)]=T
    while que:
        for j in range(len(que)):
            s = que.popleft()
            x=s[0]
            y=s[1]
            t = s[2]
            for i in range(4):
                tx = x+dx[i]
```

基本信息

#:	44838795
题目:	04115
提交人:	23n2300011503
内存:	4752kB
时间:	101ms
语言:	Python3
提交时间:	2024-05-01 00:22:37

20106: 走山路

Dijkstra, http://cs101.openjudge.cn/practice/20106/

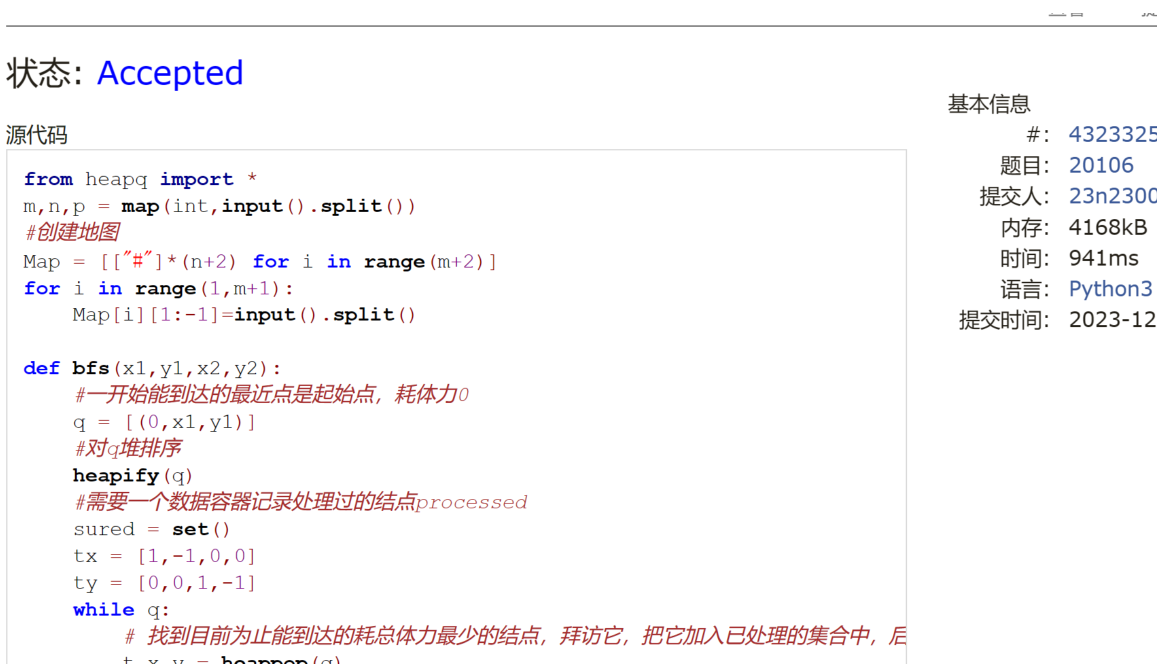
思路：

代码

```
#
from heapq import *
m,n,p = map(int,input().split())
#创建地图
```

```python
Map = [["#"]*(n+2) for i in range(m+2)]
for i in range(1,m+1):
    Map[i][1:-1]=input().split()

def bfs(x1,y1,x2,y2):
    #一开始能到达的最近点是起始点，耗体力0
    q = [(0,x1,y1)]
    #对q堆排序
    heapify(q)
    #需要一个数据容器记录处理过的结点processed
    sured = set()
    tx = [1,-1,0,0]
    ty = [0,0,1,-1]
    while q:
        # 找到目前为止能到达的耗总体力最少的结点，拜访它，把它加入已处理的集合中，后面到达此点不可能有更短的步数了
        t,x,y = heappop(q)
        sured.add((x,y))
        #如果是终点
        if x == x2 and y == y2:
            return t
        # 然后看看加入了此结点后，有哪些子结点可以到达了
        # heap的好处体现出来了，如果子节点之前就可以到达，我们不用管更新后子节点所需的体力是否下降了，因为heap会自动选出最小的
        # 比如(12,3,4) 在q中 ，现在又加入了(15,3,4)，heap在取时还是会取（12，3，4）不用我们担心
        for i in range(4):
            nx = x+tx[i]
            ny = y+ty[i]
            #不用管之前到没到过，就是要加入更新
            if Map[nx][ny]!="#" and (nx,ny) not in sured:
                heappush(q,(t+abs(int(Map[nx][ny])-int(Map[x][y])),nx,ny))
        # 现在所有的新的产生的可到达的位置又加入q中了，重复上述操作
    #如果q都空了还没return 说明根本到不了终点，因为只要能到终点，即使t很大，也最终被取出来
    return "NO"
for i in range(p):
    x1,y1,x2,y2 = map(int,input().split())
    if Map[x1+1][y1+1]=="#" or Map[x2+1][y2+1]=="#":
        print("NO")
    else:
        print(bfs(x1+1,y1+1,x2+1,y2+1))
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）


状态: Accepted

源代码

```
from heapq import *
m,n,p = map(int,input().split())
#创建地图
Map = [["#"]*(n+2) for i in range(m+2)]
for i in range(1,m+1):
    Map[i][1:-1]=input().split()

def bfs(x1,y1,x2,y2):
    #一开始能到达的最近点是起始点，耗体力0
    q = [(0,x1,y1)]
    #对q堆排序
    heapify(q)
    #需要一个数据容器记录处理过的结点processed
    sured = set()
    tx = [1,-1,0,0]
    ty = [0,0,1,-1]
    while q:
        # 找到目前为止能到达的耗总体力最少的结点，拜访它，把它加入已处理的集合中，后
        t,x,y = heappop(q)
```

基本信息

#:	4323325
题目:	20106
提交人:	23n2300
内存:	4168kB
时间:	941ms
语言:	Python3
提交时间:	2023-12

05442: 兔子与星空

Prim, http://cs101.openjudge.cn/practice/05442/

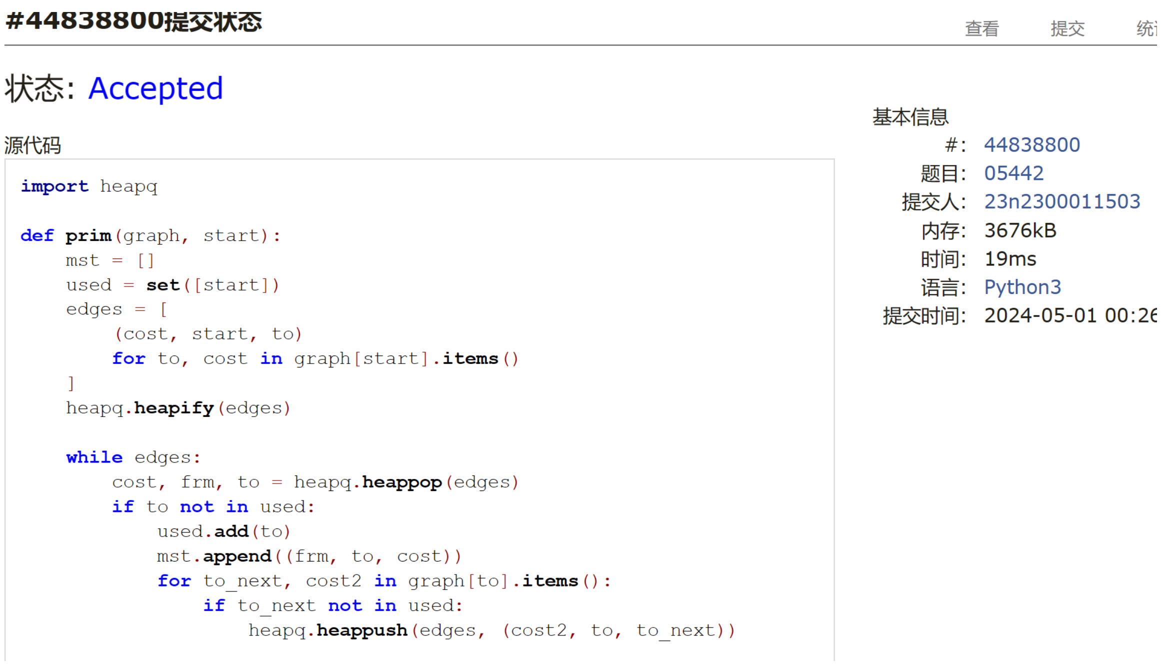
思路:

不会做，图做的太少了，多刷题

代码

```python
#
import heapq

def prim(graph, start):
    mst = []
    used = set([start])
    edges = [
        (cost, start, to)
        for to, cost in graph[start].items()
    ]
    heapq.heapify(edges)

    while edges:
        cost, frm, to = heapq.heappop(edges)
        if to not in used:
            used.add(to)
            mst.append((frm, to, cost))
            for to_next, cost2 in graph[to].items():
                if to_next not in used:
                    heapq.heappush(edges, (cost2, to, to_next))

    return mst

def solve():
    n = int(input())
    graph = {chr(i+65): {} for i in range(n)}
    for i in range(n-1):
        data = input().split()
        star = data[0]
        m = int(data[1])
        for j in range(m):
            to_star = data[2+j*2]
            cost = int(data[3+j*2])
            graph[star][to_star] = cost
            graph[to_star][star] = cost
    mst = prim(graph, 'A')
    print(sum(x[2] for x in mst))

solve()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



#44838800提交状态

查看 提交 统

状态: Accepted

源代码

```python
import heapq

def prim(graph, start):
    mst = []
    used = set([start])
    edges = [
        (cost, start, to)
        for to, cost in graph[start].items()
    ]
    heapq.heapify(edges)

    while edges:
        cost, frm, to = heapq.heappop(edges)
        if to not in used:
            used.add(to)
            mst.append((frm, to, cost))
            for to_next, cost2 in graph[to].items():
                if to_next not in used:
                    heapq.heappush(edges, (cost2, to, to_next))
```

基本信息

#:	44838800
题目:	05442
提交人:	23n2300011503
内存:	3676kB
时间:	19ms
语言:	Python3
提交时间:	2024-05-01 00:26

2. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ"2024spring每日选做"、CF、LeetCode、洛谷等网站题目。

关于带权的最值还是熟悉得多练，最后一题反复学习以下