Assignment #A: 图论：遍历，树算及栈

Updated 2018 GMT+8 Apr 21, 2024

2024 spring, Complied by 夏天明 元培学院

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

操作系统：Windows 10 | 22H2

Python编程环境：Spyder IDE 5.4.3 | Python 3.11.4 64-bit

1. 题目

20743: 整人的提词本

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

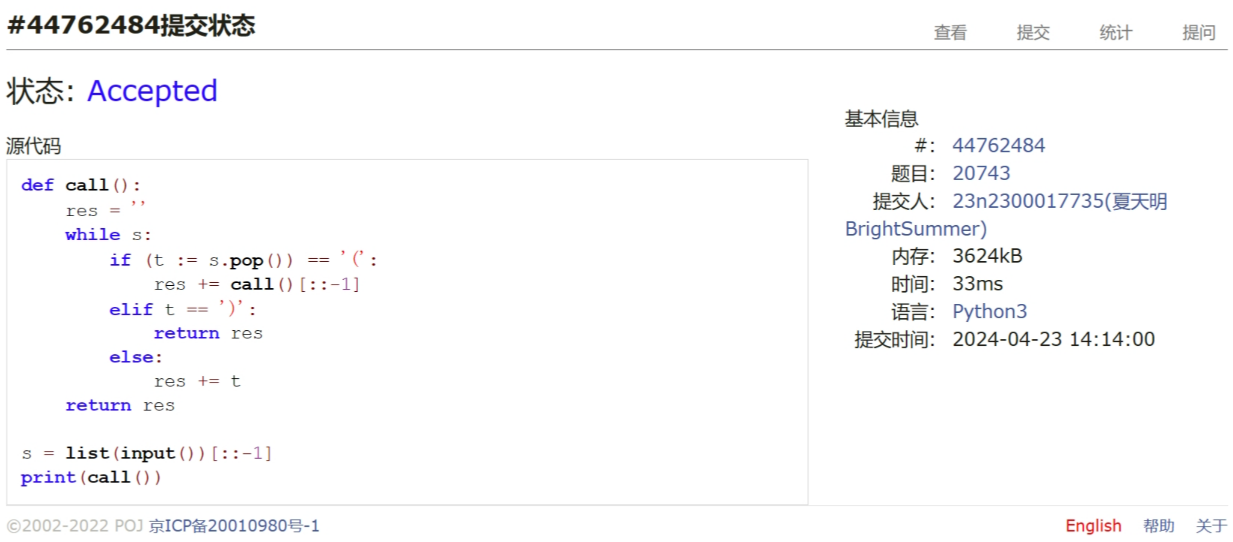
思路：使用递归处理嵌套括号问题。遇到左括号就call，遇到右括号就return

代码

```python
def call():
    res = ''
    while s:
        if (t := s.pop()) == '(':
            res += call()[::-1]
        elif t == ')':
            return res
        else:
            res += t
    return res

s = list(input())[::-1]
print(call())
```

代码运行截图



#44762484提交状态

状态: Accepted

源代码

```python
def call():
    res = ''
    while s:
        if (t := s.pop()) == '(':
            res += call()[::-1]
        elif t == ')':
            return res
        else:
            res += t
    return res

s = list(input())[::-1]
print(call())
```

基本信息

#: 44762484
题目: 20743
提交人: 23n2300017735(夏天明BrightSummer)
内存: 3624kB
时间: 33ms
语言: Python3
提交时间: 2024-04-23 14:14:00

02255: 重建二叉树

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

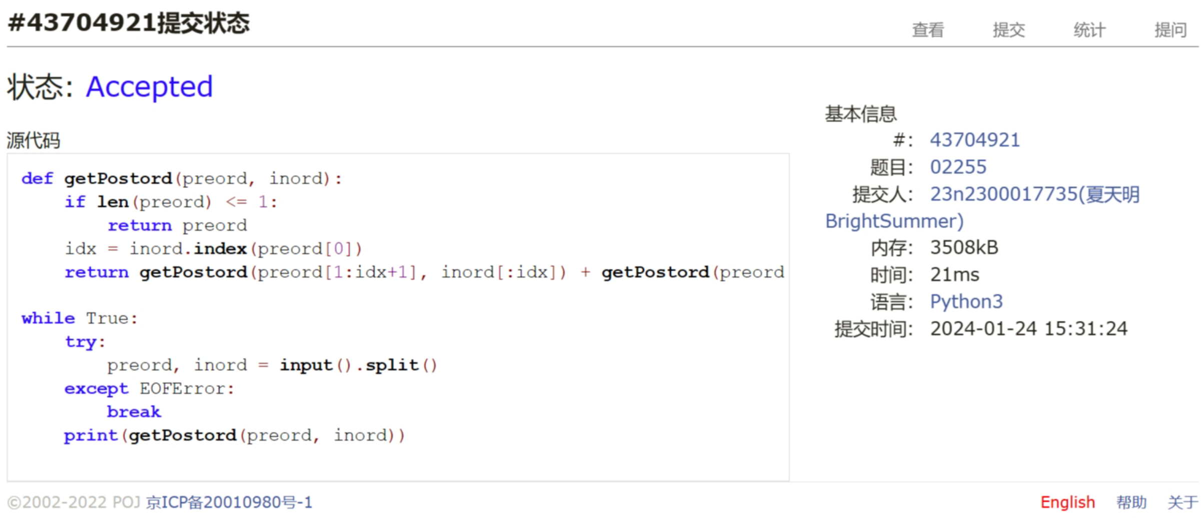
思路：直接递归

代码

```
def getPostord(preord, inord):
    if len(preord) <= 1:
        return preord
    idx = inord.index(preord[0])
    return getPostord(preord[1:idx+1], inord[:idx]) + getPostord(preord[idx+1:], 
inord[idx+1:]) + preord[0]

while True:
    try:
        preord, inord = input().split()
    except EOFError:
        break
    print(getPostord(preord, inord))
```

代码运行截图



状态: Accepted

源代码

```
def getPostord(preord, inord):
    if len(preord) <= 1:
        return preord
    idx = inord.index(preord[0])
    return getPostord(preord[1:idx+1], inord[:idx]) + getPostord(preord

while True:
    try:
        preord, inord = input().split()
    except EOFError:
        break
    print(getPostord(preord, inord))
```

基本信息
#: 43704921
题目: 02255
提交人: 23n2300017735(夏天明 BrightSummer)
内存: 3508kB
时间: 21ms
语言: Python3
提交时间: 2024-01-24 15:31:24

01426: Find The Multiple

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

要求用bfs实现

思路：先把10^k对n的余数求出来，然后从0、1开始，往前面补0、1组成不同的01串，以此类推，bfs计算余数，如果是0就找到了。注意存储visited标记扫描过的余数，下次遇到就不要重复入队了，一开始忘了导致反复MLE

用lru_cache防止反复搜索同一个n

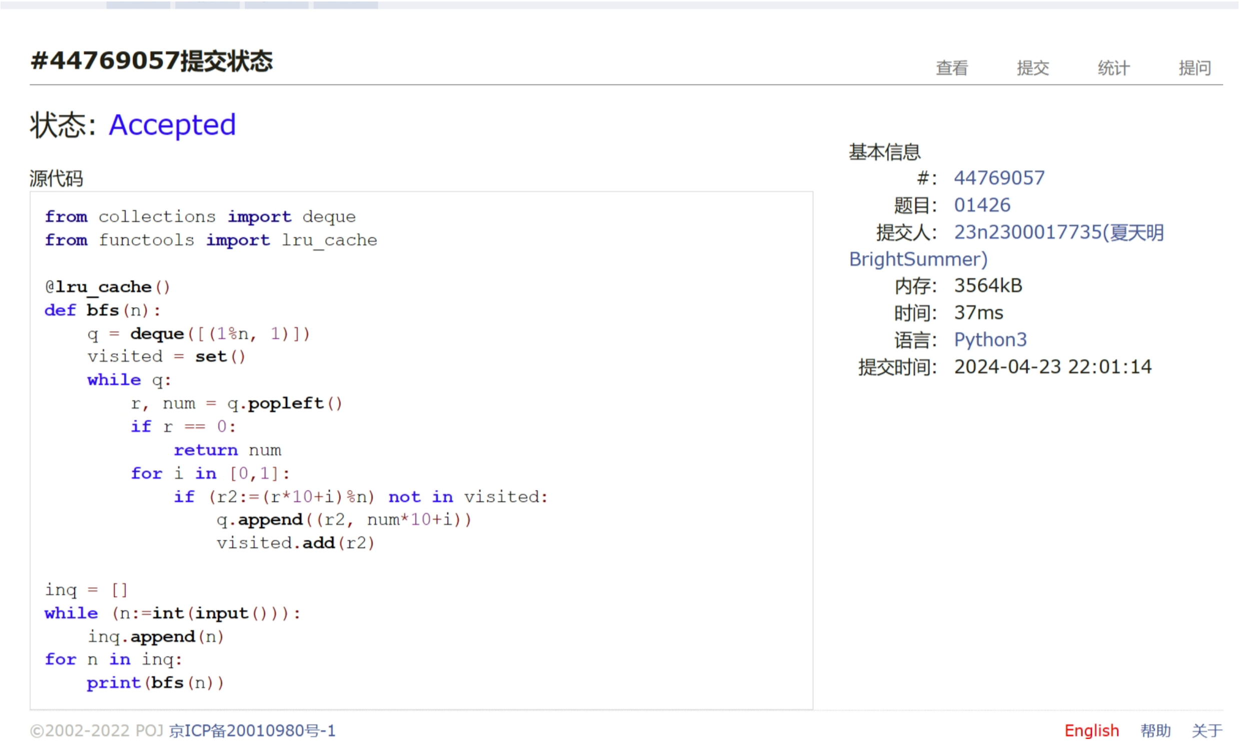
代码

```python
from collections import deque
from functools import lru_cache

@lru_cache()
def bfs(n):
    q = deque([(1%n, 1)])
    visited = set()
    while q:
        r, num = q.popleft()
        if r == 0:
            return num
        for i in [0,1]:
            if (r2:=(r*10+i)%n) not in visited:
                q.append((r2, num*10+i))
                visited.add(r2)

inq = []
while (n:=int(input())):
    inq.append(n)
for n in inq:
    print(bfs(n))
```

代码运行截图



#44769057提交状态

状态: Accepted

源代码

```python
from collections import deque
from functools import lru_cache

@lru_cache()
def bfs(n):
    q = deque([(1%n, 1)])
    visited = set()
    while q:
        r, num = q.popleft()
        if r == 0:
            return num
        for i in [0,1]:
            if (r2:=(r*10+i)%n) not in visited:
                q.append((r2, num*10+i))
                visited.add(r2)

inq = []
while (n:=int(input())):
    inq.append(n)
for n in inq:
    print(bfs(n))
```

基本信息

#:	44769057
题目:	01426
提交人:	23n2300017735(夏天明 BrightSummer)
内存:	3564kB
时间:	37ms
语言:	Python3
提交时间:	2024-04-23 22:01:14

©2002-2022 POJ 京ICP备20010980号-1

04115: 鸣人和佐助

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

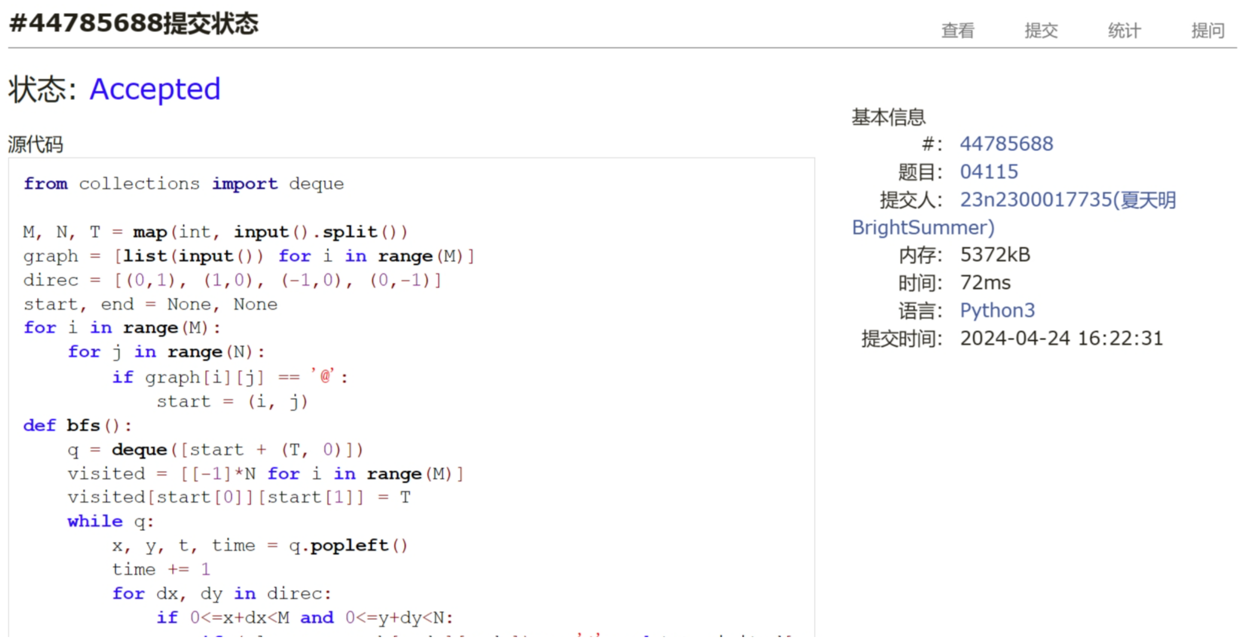
思路：稍复杂的bfs问题，visited需要维护经过时的最大查克拉数

代码

```python
from collections import deque

M, N, T = map(int, input().split())
graph = [list(input()) for i in range(M)]
direc = [(0,1), (1,0), (-1,0), (0,-1)]
start, end = None, None
for i in range(M):
    for j in range(N):
        if graph[i][j] == '@':
            start = (i, j)
def bfs():
    q = deque([start + (T, 0)])
    visited = [[-1]*N for i in range(M)]
    visited[start[0]][start[1]] = T
    while q:
        x, y, t, time = q.popleft()
        time += 1
        for dx, dy in direc:
            if 0<=x+dx<M and 0<=y+dy<N:
                if (elem := graph[x+dx][y+dy]) == '*' and t > visited[x+dx][y+dy]:
                    visited[x+dx][y+dy] = t
                    q.append((x+dx, y+dy, t, time))
                elif elem == '#' and t > 0 and t-1 > visited[x+dx][y+dy]:
                    visited[x+dx][y+dy] = t-1
                    q.append((x+dx, y+dy, t-1, time))
                elif elem == '+':
                    return time
    return -1
print(bfs())
```

代码运行截图



#44785688提交状态

状态: Accepted

源代码

```
from collections import deque

M, N, T = map(int, input().split())
graph = [list(input()) for i in range(M)]
direc = [(0,1), (1,0), (-1,0), (0,-1)]
start, end = None, None
for i in range(M):
    for j in range(N):
        if graph[i][j] == '@':
            start = (i, j)
def bfs():
    q = deque([start + (T, 0)])
    visited = [[-1]*N for i in range(M)]
    visited[start[0]][start[1]] = T
    while q:
        x, y, t, time = q.popleft()
        time += 1
        for dx, dy in direc:
            if 0<=x+dx<M and 0<=y+dy<N:
```

基本信息
#: 44785688
题目: 04115
提交人: 23n2300017735(夏天明BrightSummer)
内存: 5372kB
时间: 72ms
语言: Python3
提交时间: 2024-04-24 16:22:31

20106: 走山路

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

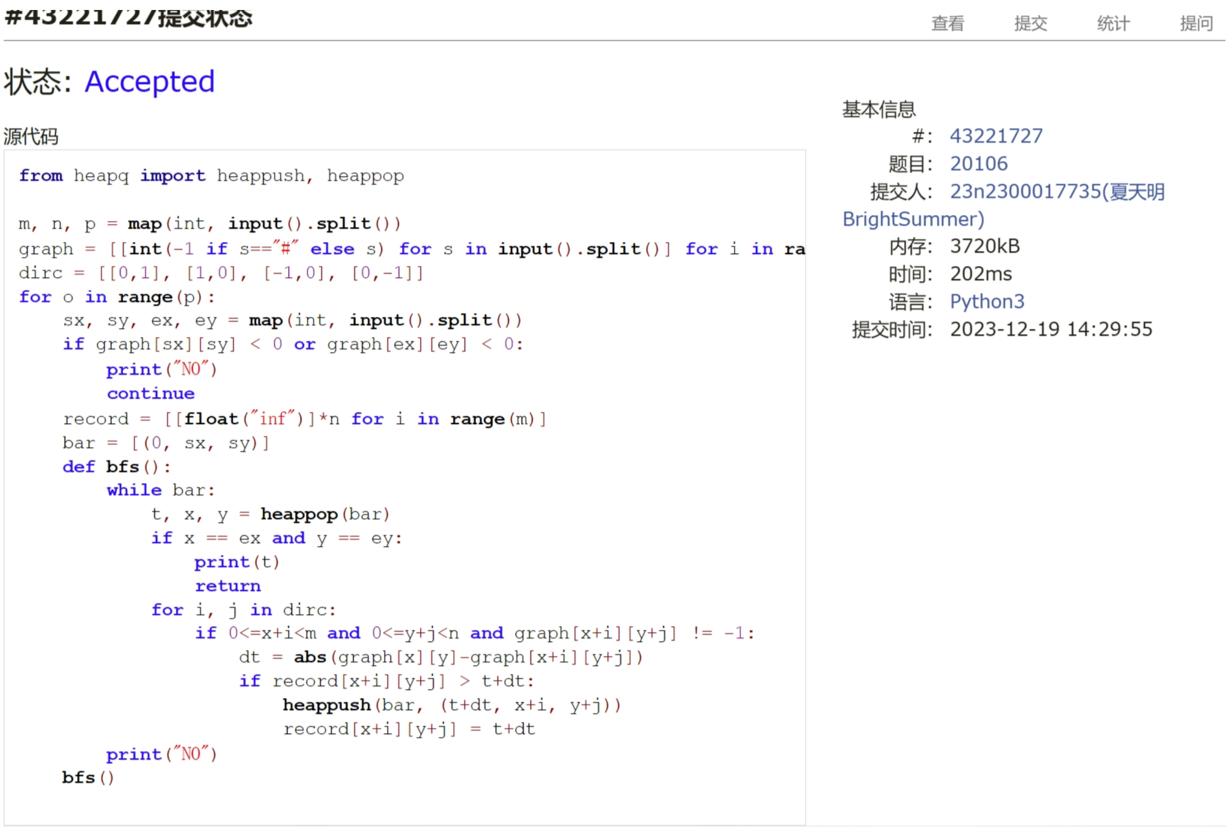
思路：直接dijkstra

代码

```
from heapq import heappush, heappop

m, n, p = map(int, input().split())
graph = [[int(-1 if s=="#" else s) for s in input().split()] for i in range(m)]
dirc = [[0,1], [1,0], [-1,0], [0,-1]]
for o in range(p):
    sx, sy, ex, ey = map(int, input().split())
    if graph[sx][sy] < 0 or graph[ex][ey] < 0:
        print("NO")
        continue
    record = [[float("inf")]*n for i in range(m)]
    bar = [(0, sx, sy)]
    def bfs():
        while bar:
            t, x, y = heappop(bar)
            if x == ex and y == ey:
                print(t)
                return
            for i, j in dirc:
                if 0<=x+i<m and 0<=y+j<n and graph[x+i][y+j] != -1:
                    dt = abs(graph[x][y]-graph[x+i][y+j])
                    if record[x+i][y+j] > t+dt:
                        heappush(bar, (t+dt, x+i, y+j))
                        record[x+i][y+j] = t+dt
        print("NO")
    bfs()
```

代码运行截图



#43221727提交状态

状态: Accepted

源代码

```
from heapq import heappush, heappop

m, n, p = map(int, input().split())
graph = [[int(-1 if s=="#" else s) for s in input().split()] for i in ra
dirc = [[0,1], [1,0], [-1,0], [0,-1]]
for o in range(p):
    sx, sy, ex, ey = map(int, input().split())
    if graph[sx][sy] < 0 or graph[ex][ey] < 0:
        print("NO")
        continue
    record = [[float("inf")]*n for i in range(m)]
    bar = [(0, sx, sy)]
    def bfs():
        while bar:
            t, x, y = heappop(bar)
            if x == ex and y == ey:
                print(t)
                return
            for i, j in dirc:
                if 0<=x+i<m and 0<=y+j<n and graph[x+i][y+j] != -1:
                    dt = abs(graph[x][y]-graph[x+i][y+j])
                    if record[x+i][y+j] > t+dt:
                        heappush(bar, (t+dt, x+i, y+j))
                        record[x+i][y+j] = t+dt
        print("NO")
    bfs()
```

基本信息

#:	43221727
题目:	20106
提交人:	23n2300017735(夏天明 BrightSummer)
内存:	3720kB
时间:	202ms
语言:	Python3
提交时间:	2023-12-19 14:29:55

05442: 兔子与星空

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

思路：最小生成树的Prim算法

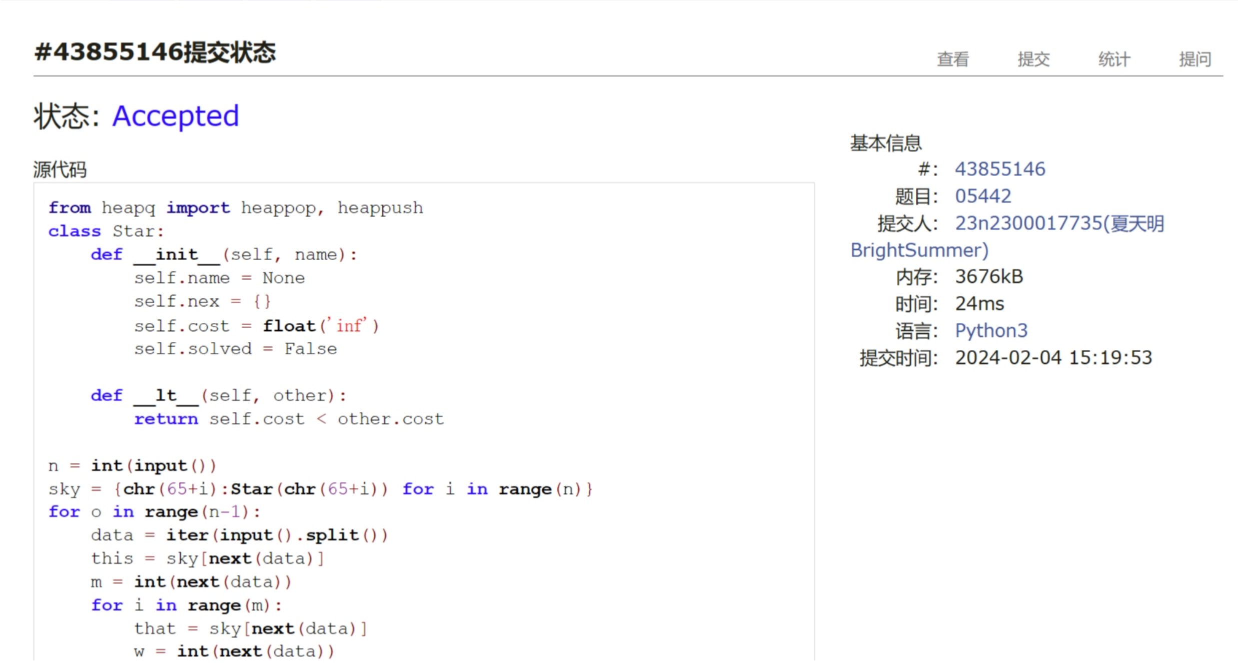
代码

```
from heapq import heappop, heappush
class Star:
    def __init__(self, name):
        self.name = None
        self.nex = {}
        self.cost = float('inf')
        self.solved = False
```

```python
    def __lt__(self, other):
        return self.cost < other.cost

n = int(input())
sky = {chr(65+i):Star(chr(65+i)) for i in range(n)}
for o in range(n-1):
    data = iter(input().split())
    this = sky[next(data)]
    m = int(next(data))
    for i in range(m):
        that = sky[next(data)]
        w = int(next(data))
        this.nex[that] = that.nex[this] = w
q = [sky['A']]
sky['A'].cost = 0
ans = 0
while n:
    star = heappop(q)
    if star.solved:
        continue
    else:
        star.solved = True
        ans += star.cost
        n -= 1
    for new, w in star.nex.items():
        if not new.solved and w < new.cost:
            new.cost = w
            heappush(q, new)
print(ans)
```

代码运行截图



#43855146提交状态

状态: Accepted

源代码

```python
from heapq import heappop, heappush
class Star:
    def __init__(self, name):
        self.name = None
        self.nex = {}
        self.cost = float('inf')
        self.solved = False

    def __lt__(self, other):
        return self.cost < other.cost

n = int(input())
sky = {chr(65+i):Star(chr(65+i)) for i in range(n)}
for o in range(n-1):
    data = iter(input().split())
    this = sky[next(data)]
    m = int(next(data))
    for i in range(m):
        that = sky[next(data)]
        w = int(next(data))
```

基本信息

#: 43855146
题目: 05442
提交人: 23n2300017735(夏天明BrightSummer)
内存: 3676kB
时间: 24ms
语言: Python3
提交时间: 2024-02-04 15:19:53

2. 学习总结和收获

复习了图的一些较为复杂的算法，包括上学期学的带有附加条件的bfs。以及Dijk和Prim算法，都属于特殊的bfs，维护堆实现不同的遍历顺序