Assignment #A: 图论：算法，树算及栈

Updated 2018 GMT+8 Apr 21, 2024

2024 spring, Complied by ==狄晨阳 生命科学学院==

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

==（请改为同学的操作系统、编程环境等）==

操作系统：Windows11

Python编程环境：Spyder IDE 5.4.3

C/C++编程环境：无

1. 题目

20743: 整人的提词本

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

思路：使用一个栈来处理输入，再用一个缓存来记录每次的反转结果并将其重新入栈

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Tue Apr 30 16:52:58 2024

@author: 20311
"""

s=input().strip()
stack=[]
stack2=[]
for char in s:
    if char!=')':
```

```
        stack.append(char)
    else:
        while stack and stack[-1]!='(':
            a=stack.pop()
            stack2.append(a)
        if stack:
            stack.pop()

        while stack2:
            a=stack2.pop(0)
            stack.append(a)

print(''.join(stack))
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 30 16:52:58 2024

@author: 20311
"""

s=input().strip()
stack=[]
stack2=[]
for char in s:
    if char!=')':
        stack.append(char)
    else:
        while stack and stack[-1]!='(':
            a=stack.pop()
            stack2.append(a)
        if stack:
            stack.pop()

        while stack2:
            a=stack2.pop(0)
            stack.append(a)

print(''.join(stack))
```

基本信息

#:	44836624
题目:	20743
提交人:	23n2300012138(yukino)
内存:	3592kB
时间:	21ms
语言:	Python3
提交时间:	2024-04-30 17:01:01

02255: 重建二叉树

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

思路：多组根据前中序输出后序

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Tue Apr 30 17:14:13 2024
```

```
@author: 20311
"""

def build(pre,mid):
    if not pre or not mid:
        return []
    root=pre[0]
    l=mid.index(root)

    pre_left=pre[1:l+1]
    pre_right=pre[l+1:]

    mid_left=mid[:l]
    mid_right=mid[l+1:]

    ans=[]
    ans+=build(pre_left,mid_left)
    ans+=build(pre_right,mid_right)
    ans+=[root]

    return ans

while True:
    try:
        pre,mid=input().split()
        pre=list(pre)
        mid=list(mid)
        ans=build(pre,mid)
        print(''.join(ans))
    except EOFError:
        break
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 30 17:14:13 2024

@author: 20311
"""

def build(pre,mid):
    if not pre or not mid:
        return []
    root=pre[0]
    l=mid.index(root)

    pre_left=pre[1:l+1]
    pre_right=pre[l+1:]

    mid_left=mid[:l]
    mid_right=mid[l+1:]

    ans=[]
    ans+=build(pre_left,mid_left)
    ans+=build(pre_right,mid_right)
    ans+=[root]

    return ans

while True:
    try:
        pre,mid=input().split()
        pre=list(pre)
        mid=list(mid)
        ans=build(pre,mid)
        print(''.join(ans))
    except EOFError:
        break
```

基本信息

#:	44837088
题目:	02255
提交人:	23n2300012138(yukino)
内存:	3528kB
时间:	20ms
语言:	Python3
提交时间:	2024-04-30 18:26:25

01426: Find The Multiple

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

要求用bfs实现

思路：一开始使用穷举超时了，后来使用了尝试所有0和1的组合，还是超时，最后看了题解利用模来舍去一些不必要的结果

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Tue Apr 30 19:47:43 2024

@author: 20311
"""
from collections import deque
def search(n):
    q=deque()
    q.append((1,'1'))
    visited={1}

    while q:
        mod,num=q.popleft()
```

```
        if mod==0:
            return num

        for i in [0,1]:
            new_mod=(mod*10+i)%n
            if new_mod not in visited:
                visited.add(new_mod)
                q.append((new_mod,num+str(i)))

while True:
    n=int(input())
    if n==0:
        break

    print(search(n))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 30 19:47:43 2024

@author: 20311
"""
from collections import deque
def search(n):
    q=deque()
    q.append((1,'1'))
    visited={1}

    while q:
        mod,num=q.popleft()
        if mod==0:
            return num

        for i in [0,1]:
            new_mod=(mod*10+i)%n
            if new_mod not in visited:
                visited.add(new_mod)
                q.append((new_mod,num+str(i)))

while True:
    n=int(input())
    if n==0:
        break

    print(search(n))
```

基本信息

#: 44837743
题目: 01426
提交人: 23n2300012138(yukino)
内存: 3592kB
时间: 38ms
语言: Python3
提交时间: 2024-04-30 20:25:02

04115: 鸣人和佐助

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

思路：典型的bfs，增加了一个判据

代码

```python
# # -*- coding: utf-8 -*-
"""
Created on Tue Apr 30 20:42:36 2024

@author: 20311
"""
from collections import deque

m,n,t=map(int,input().split())
maap=[]
for i in range(m):
    a=list(input())
    for j in range(n):
        if a[j]=='@':
            start=(i,j,t,0)
    maap.append(a)

nei=[(-1, 0), (0, -1), (1, 0), (0, 1)]
visited=set()
i,j,t,s=start
visited.add((i,j,t))
q=deque()
q.append(start)
jg=False
while q:
    i,j,c,s=q.popleft()
    s+=1
    for di,dj in nei:
        x=i+di
        y=j+dj
        if 0<=x<m and 0<=y<n:
            nc=c
            if maap[x][y]=='#':
                nc=c-1
            if maap[x][y]=='+' and nc>=0:
                jg=True
                print(s)
                break

            if nc>=0 and (x,y,nc) not in visited:
                q.append((x,y,nc,s))
                visited.add((x,y,nc))
    if jg:
        break
if not jg:
    print(-1)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

基本信息

#: 44838102
题目: 04115
提交人: 23n2300012138(yukino)
内存: 7572kB
时间: 101ms
语言: Python3
提交时间: 2024-04-30 21:29:41

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 30 20:42:36 2024

@author: 20311
"""
from collections import deque

m,n,t=map(int,input().split())
maap=[]
for i in range(m):
    a=list(input())
    for j in range(n):
        if a[j]=='@':
            start=(i,j,t,0)
    maap.append(a)

nei=[(-1, 0), (0, -1), (1, 0), (0, 1)]
visited=set()
i,j,t,s=start
visited.add((i,j,t))
q=deque()
q.append(start)
jg=False
while q:
    i,j,c,s=q.popleft()
    s+=1
    for di,dj in nei:
        x=i+di
        y=j+dj
        if 0<=x<m and 0<=y<n:
            nc=c
            if maap[x][y]=='#':
                nc=c-1
            if maap[x][y]=='+' and nc>=0:
                jg=True
                print(s)
                break

            if nc>=0 and (x,y,nc) not in visited:
                q.append((x,y,nc,s))
                visited.add((x,y,nc))
    if jg:
        break
if not jg:
    print(-1)
```

20106: 走山路

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

思路：使用heapq来缩短时间，保证每次都是最优的

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Tue Apr 30 21:31:14 2024

@author: 20311
"""
from heapq import heappush,heappop
m,n,p=map(int,input().split())
maap=[]
nei=[(0,1),(0,-1),(1,0),(-1,0)]
```

```
for _ in range(m):
    maap.append(input().split())

for _ in range(p):
    jg=False
    x1,y1,x2,y2=map(int,input().split())
    if maap[x1][y1]=='#' or maap[x2][y2]=="#":
        print('NO')
        continue
    visited=set()
    q=[(0,x1,y1)]

    while q:
        t,x,y=heappop(q)
        if (x,y) in visited:
            continue
        visited.add((x,y))
        if x==x2 and y==y2:
            jg=True
            print(t)
            break
        for dx,dy in nei:
            nx=x+dx
            ny=y+dy
            if 0<=nx<m and 0<=ny<n and maap[nx][ny]!='#' and (nx,ny) not in
visited:
                nt=t+abs(int(maap[nx][ny])-int(maap[x][y]))
                heappush(q,(nt,nx,ny))
    if not jg:
        print('NO')
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态：Accepted

基本信息

#：44838410
题目：20106
提交人：23n2300012138(yukino)
内存：3916kB
时间：254ms
语言：Python3
提交时间：2024-04-30 22:23:53

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 30 21:31:14 2024

@author: 20311
"""
from heapq import heappush,heappop
m,n,p=map(int,input().split())
maap=[]
nei=[(0,1),(0,-1),(1,0),(-1,0)]
for _ in range(m):
    maap.append(input().split())

for _ in range(p):
    jg=False
    x1,y1,x2,y2=map(int,input().split())
    if maap[x1][y1]=='#' or maap[x2][y2]=="#":
        print('NO')
        continue
    visited=set()
    q=[(0,x1,y1)]

    while q:
        t,x,y=heappop(q)
        if (x,y) in visited:
            continue
        visited.add((x,y))
        if x==x2 and y==y2:
            jg=True
            print(t)
            break
        for dx,dy in nei:
            nx=x+dx
            ny=y+dy
            if 0<=nx<m and 0<=ny<n and maap[nx][ny]!='#' and (nx,ny) not
                nt=t+abs(int(maap[nx][ny])-int(maap[x][y]))
                heappush(q,(nt,nx,ny))
    if not jg:
        print('NO')
```

05442: 兔子与星空

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

思路：用嵌套字典来建图，然后以权重建堆，找出维系最基本联系的最小权重

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Tue Apr 30 22:29:15 2024

@author: 20311
"""

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"）==

状态：Accepted

源代码

```python
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 30 22:29:15 2024

@author: 20311
"""

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()
```

基本信息

#:	44838453
题目:	05442
提交人:	23n2300012138(yukino)
内存:	3680kB
时间:	21ms
语言:	Python3
提交时间:	2024-04-30 22:35:22

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。==

是

本次着重复习了bfs，最后两道题都是学习了一下题解，用heapq中的函数来减少了时间并优化了算法，需要进一步强化练习