Assignment #A: 图论：算法，树算及栈

Updated 2018 GMT+8 Apr 21, 2024

2024 spring, Complied by 赵语涵 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

操作系统：windows 11

Python编程环境：Spyder IDE 5.2.2

1. 题目

20743: 整人的提词本

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

思路：看到括号相关便很容易想到用栈的方法。将字符存入栈，在遇到有括号时依次pop出直到左括号，然后将弹出的字符串（实现了倒序）再放回原来的栈中即可。

代码

```
#赵语涵2300012254
temp = []
for i in list(input()):
    if i != ')':
        temp.append(i)
    else:
        flop = []
        while (x:=temp.pop()) != '(':
            flop.append(x)
        temp += flop
print(''.join(temp))
```

代码运行截图

#44937978提交状态

查看 提交 统计

状态: Accepted

源代码

```
#赵语涵2300012254
temp = []
for i in list(input()):
    if i != ')':
        temp.append(i)
    else:
```

基本信息

#: 44937978
题目: 20743
提交人: 23n2300012254
内存: 3600kB
时间: 25ms
语言: Python3
提交时间: 2024-05-12 13:39:02

02255: 重建二叉树

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

思路：总感觉以前用很麻烦的方法做过这道题但是OJ上显示没有提交记录...通过以前建二叉树的经验很快想到了用tree函数递归，返回tree.left+tree.right+tree.value即得到后序序列，在条件和具体实施上花了一些时间，最后代码感觉比以前简洁精炼很多

代码

```
#赵语涵2300012254
def tree(lis,l_left,l_right):
    global index
    if not lis:
        return ''
    x = lis[0]
    i = index[x]
    tree_left,tree_right = '',''
    if i > l_left+1:
        tree_left = tree(lis[1:i-l_left],l_left,i)
    if i<l_right-1:
        tree_right = tree(lis[i-l_left:],i,l_right)
    return tree_left+tree_right+x
while True:
    try:
        pre,mid = input().split()
        index = dict(zip(list(mid),range((l:=len(pre)))))
        answer = tree(list(pre),-1,l)
        print(answer)
    except EOFError:
        break
```

代码运行截图

#44947856提交状态 查看 提交 统计

状态: Accepted

源代码

```
#赵语涵2300012254
def tree(lis,l_left,l_right):
    global index
    if not lis:
        return ''
    x = lis[0]
    i = index[x]
```

基本信息

#: 44947856
题目: 02255
提交人: 23n2300012254
内存: 7428kB
时间: 32ms
语言: Python3
提交时间: 2024-05-12 22:19:07

01426: Find The Multiple

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

要求用bfs实现

思路：用bfs的思路很快AC了

代码

```python
#赵语涵2300012254
from collections import deque
def bfs():
    lis = deque(['1'])
    while True:
        x = lis.popleft()
        if int(x) % n == 0:
            return x
        lis.append(x+'0')
        lis.append(x+'1')
while True:
    n = int(input())
    if n == 0:
        break
    print(bfs())
```

代码运行截图

#44948360提交状态 查看 提交 统计

状态: Accepted

源代码

```
#赵语涵2300012254
from collections import deque
def bfs():
    lis = deque(['1'])
    while True:
        x = lis.popleft()
```

基本信息

#: 44948360
题目: 01426
提交人: 23n2300012254
内存: 49888kB
时间: 836ms
语言: Python3
提交时间: 2024-05-12 23:58:56

04115: 鸣人和佐助

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

思路：最开始用dfs做超时了，然后用bfs的方法，与常规bfs不同的是多了参数t限制鸣人是否能通过敌人。设置了chark存储某个位置当前最大的查克拉数，只有1）当前c>最大查克拉且下一位置是'#'或2）当前位置c-1>最大查克拉且下一位置是'*'时更新进入下一位置

关于dfs和bfs：需要找最小路径时用bfs，需要遍历所有路径（比如总结有几条通路）时用dfs

代码

```
#赵语涵2300012254
from collections import deque
directions = [(-1,0),(0,1),(1,0),(0,-1)]
def bfs():
    global chark,answer
    while pos:
        x = pos.popleft()
        c = x[2]
        for d in directions:
            if 0<=(a:=x[0]+d[0])<m and 0<=(b:=x[1]+d[1])<n:
                if fig[a][b] == '+':
                    answer = x[3]+1
                    return
                if fig[a][b]=='*' and c>chark[(a,b)]:
                    chark[(a,b)] = c
                    pos.append((a,b,c,x[3]+1))
                elif c>0:
                    if c-1>chark[(a,b)]:
                        pos.append((a,b,c-1,x[3]+1))
                        chark[(a,b)] = c-1

m,n,t = map(int,input().split())
fig,chark = [],{}
for i in range(m):
    fig.append(input())
    for j in range(n):
        chark[(i,j)] = -1
        if fig[i][j] == '@':
            start_x,start_y = i,j
            chark[(i,j)] = t
pos,answer = deque([(start_x,start_y,t,0)]),-1
bfs()
print(answer)
```

代码运行截图

#44958711提交状态 查看 提交 统计

状态: Accepted

源代码

```
#赵语涵2300012254
from collections import deque
directions = [(-1,0),(0,1),(1,0),(0,-1)]
def bfs():
    global chark,answer
    while pos:
        x = pos.popleft()
```

基本信息

#: 44958711
题目: 04115
提交人: 23n2300012254
内存: 6924kB
时间: 83ms
语言: Python3
提交时间: 2024-05-14 11:33:05

20106: 走山路

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

思路：借的以前的代码、、

使用dijkstra，本来按照算法图解上的模板写，发现其实它有很多可以改进的地方，尤其对于这道题有几组数据，每次都要找最短路径的情况下就会超时

通过上学期的cheating paper再次复习了dijkstra算法的思路与模式代码

代码

```
# -*- coding: utf-8 -*-
"""
Created on Thu Dec 21 15:35:09 2023

@author: 赵语涵2300012254
"""

import heapq

m,n,p = map(int,input().split())
plot = [list(input().split()) for _ in range(m)]

def differ(x,y):
    return abs(int(x)-int(y))

for _ in range(p):
    costs,processed = [],set()
    a,b,c,d = map(int,input().split())
    if plot[a][b]=='#' or plot[c][d]=='#':
        print('NO')
        continue
    sta,stop = (a,b),(c,d)
    heapq.heappush(costs, (0,a,b))
    answer = 'NO'
    while costs:
        co,i,j = heapq.heappop(costs)
        if (i,j) == stop:
```

```
            answer = co
            break
        for x,y in zip([i-1,i,i,i+1],[j,j-1,j+1,j]):
            if 0<=x<=m-1 and 0<=y<=n-1:
                neighbor = (x,y)
                if neighbor not in processed and plot[x][y]!='#':
                    cost = co+differ(plot[x][y],plot[i][j])
                    heapq.heappush(costs, (cost,x,y))
        processed.add((i,j))
    print(answer)
```

代码运行截图

#43283248提交状态 查看 提交 统计

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Thu Dec 21 15:35:09 2023

@author: 赵语涵2300012254
"""
```

基本信息

#: 43283248
题目: 20106
提交人: 23n2300012254
内存: 4144kB
时间: 1495ms
语言: Python3
提交时间: 2023-12-21 23:45:21

05442: 兔子与星空

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

思路：感觉Kruskal算法比较简洁，但是实际使用时发现最大的问题出现在并查集的构建，通过题解加深了理解。

用heap依次弹出当前权值最小的边并用并查集判断该边加入后是否会出现连通域的状况（两个端点不能是共同祖先）

代码

```
#赵语涵2300012254
import heapq                                        #处理数据输入
edges = [];heapq.heapify(edges)
n = int(input())
for _ in range(n-1):
    data = input()
    if data[2] == 0:                        #字符串第3位为0时没有连接之后的节点
        continue
    else:
        process = data.split()
        for j in range(int(process[1])):
            node,weight = process[2*j+2],int(process[2*j+3])
            heapq.heappush(edges, (weight,data[0],node))
```

```python

parent = {chr(x):chr(x) for x in range(65,91)}     #建立并查集
def get_root(x):                                   #递归得到祖先
    if x == parent[x]:
        return x
    return get_root(parent[x])
ancient = 0
def merge(a,b):                                    #祖先相同说明边已连通，不同则将祖先1指向祖先2
    global ancient
    if (p_1:=get_root(a)) != (p_2:=get_root(b)):
        if p_1 in ancient:
            parent[p_2] = p_1
            ancient -= {p_2}
        elif p_2 in ancient:
            parent[p_1] = p_2
            ancient -= {p_1}
        else:
            parent[p_1] = p_2
            ancient.add(p_2)
        return True
    else:
        return False

counted = {chr(x):False for x in range(65,91)}      #从heap中弹出最小边，直到所有节点都已连通
weights,count,ancient = 0,0,set()
while True:
    if count == n and len(ancient) == 1:
        break                                      #当总结点数=n时停止循环输出当前总权重
    edge = heapq.heappop(edges)
    node_1,node_2 = edge[1],edge[2]
    if merge(node_1, node_2):      #仅当祖先不同时加入当前最小边
        for i in [node_1,node_2]:     #若某节点 没有计入，总节点数count+1
            if not counted[i]:
                count += 1
                counted[i] = True
        weights += edge[0]             #权重加入该最小边
print(weights)
```

代码运行截图

#44959772提交状态

查看　提交　统计

状态: Accepted

源代码

```python
#赵语涵2300012254
import heapq                                       #处理数据输入
edges = [];heapq.heapify(edges)
n = int(input())
for _ in range(n-1):
    data = input()
```

基本信息

#: 44959772
题目: 05442
提交人: 23n2300012254
内存: 3732kB
时间: 22ms
语言: Python3
提交时间: 2024-05-14 14:10:55

2. 学习总结和收获

感觉通过练习，对dfs和bfs的使用条件和代码书写都有更好的总结，以及学习了最小生成树的两种算法（Prim和Kruskal;另外学习练习了并查集的使用