Assignment #A: 图论：遍历，树算及栈

Updated 2018 GMT+8 Apr 21, 2024

2024 spring, Complied by 王业成 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：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

```
# s=input().strip()
stack=[]
for i in s:
    if i!=")":
        stack.append(i)
    else:
        a=[]
        while stack and stack[-1]!="(":
            a.append(stack.pop())
        stack.pop()
        stack.extend(a)
print("".join(stack))
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
s=input().strip()
stack=[]
for i in s:
    if i!=")":
        stack.append(i)
    else:
        a=[]
        while stack and stack[-1]!="(":
            a.append(stack.pop())
        stack.pop()
        stack.extend(a)
print("".join(stack))
```

基本信息

#: 44836123
题目: 20743
提交人: wangyecheng
内存: 3596kB
时间: 20ms
语言: Python3
提交时间: 2024-04-30 15:18:59

02255: 重建二叉树

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

思路:

代码

```
#
def buildtree(preorder,inorder):
    if not preorder:
        return ""
    root=preorder[0]
    i=inorder.index(root)
    left_preorder=preorder[1:1+i]
    right_preorder=preorder[1+i:]
    left_inorder=inorder[:i]
    right_inorder=inorder[i+1:]
    left_tree=buildtree(left_preorder,left_inorder)
    right_tree=buildtree(right_preorder,right_inorder)
    return left_tree+right_tree+root
```

```
while True:
    try:
        preorder,inorder=input().split()
        print(buildtree(preorder,inorder))
    except EOFError:
        break
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
def buildtree(preorder,inorder):
    if not preorder:
        return ""
    root=preorder[0]
    i=inorder.index(root)
    left_preorder=preorder[1:1+i]
    right_preorder=preorder[1+i:]
    left_inorder=inorder[:i]
    right_inorder=inorder[i+1:]
    left_tree=buildtree(left_preorder,left_inorder)
    right_tree=buildtree(right_preorder,right_inorder)
    return left_tree+right_tree+root
while True:
    try:
        preorder,inorder=input().split()
        print(buildtree(preorder,inorder))
    except EOFError:
        break
```

基本信息

#:	44836190
题目:	02255
提交人:	wangyecheng
内存:	3516kB
时间:	20ms
语言:	Python3
提交时间:	2024-04-30 15:39:31

01426: Find The Multiple

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

要求用bfs实现

思路：利用队列bfs实现

代码

```
#
from collections import deque
def find(n):
    q=deque()
    q.append((1%n,"1"))
    visited=set([1%n])
    while q:
        mod,num=q.popleft()
        if mod==0:
            return num
        else:
            for i in ["0","1"]:
                mod=(mod*10+int(i))%n
                num=num+i
                if mod not in visited:
```

```
                    q.append((mod,num))
                    visited.add(mod)
while True:
    n=int(input())
    if n!=0:
        print(find(n))
    else:
        break
```

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
def find(n):
    q=deque()
    q.append((1%n,"1"))
    visited=set([1%n])
    while q:
        mod,num=q.popleft()
        if mod==0:
            return num
        else:
            for i in ["0","1"]:
                mod=(mod*10+int(i))%n
                num=num+i
                if mod not in visited:
                    q.append((mod,num))
                    visited.add(mod)
while True:
    n=int(input())
    if n!=0:
        print(find(n))
    else:
        break
```

基本信息

#: 44836788
题目: 01426
提交人: wangyecheng
内存: 3636kB
时间: 41ms
语言: Python3
提交时间: 2024-04-30 17:26:20

04115: 鸣人和佐助

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

思路:

代码

```
#
from collections import deque
m,n,t=map(int,input().split())
dirs=[(1,0),(-1,0),(0,1),(0,-1)]
chess=[list(input()) for i in range(m)]
for i in range(m):
    for j in range(n):
        if chess[i][j]=="@":
            start=(i,j)
def bfs(t):
    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 dirs:
            if 0 <= x + dx < m and 0 <= y + dy < n:
                if chess[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 chess[x + dx][y + dy] == '#' 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 chess[x + dx][y + dy] == '+':
                    return time
    return -1
step=bfs(t)
print(step)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

基本信息

#: 44837417
题目: 04115
提交人: wangyecheng
内存: 4104kB
时间: 75ms
语言: Python3
提交时间: 2024-04-30 19:31:42

源代码

```
from collections import deque
m,n,t=map(int,input().split())
dirs=[(1,0),(-1,0),(0,1),(0,-1)]
chess=[list(input()) for i in range(m)]
for i in range(m):
    for j in range(n):
        if chess[i][j]=="@":
            start=(i,j)
def bfs(t):
    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 dirs:
            if 0 <= x + dx < m and 0 <= y + dy < n:
                if chess[x + dx][y + dy] == '*' and t > visited[x + dx][
                    visited[x + dx][y + dy] = t
                    q.append((x + dx, y + dy, t, time))
                elif chess[x + dx][y + dy] == '#' and t > 0 and t - 1 >
                    visited[x + dx][y + dy] = t - 1
                    q.append((x + dx, y + dy, t - 1, time))
                elif chess[x + dx][y + dy] == '+':
                    return time
    return -1
step=bfs(t)
print(step)
```

20106: 走山路

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

思路:

代码

```
#
import heapq
m,n,t=map(int,input().split())
dirs=[(1,0),(-1,0),(0,1),(0,-1)]
chess=[list(input().split()) for i in range(m)]
for _ in range(t):
    sx, sy, ex, ey = map(int, input().split())
    if chess[sx][sy] == "#" or chess[ex][ey] == "#":
        print("NO")
        continue
    visited, heap,step= set(), [], []
    heapq.heappush(heap, (0, sx, sy))
    visited.add((sx, sy, -1))
    while heap:
        tire, x, y = heapq.heappop(heap)
        if x == ex and y == ey:
            step.append(tire)
        for i in range(4):
            dx, dy = dirs[i]
            x1, y1 = dx+x, dy+y
            if 0 <= x1 < m and 0 <= y1 < n and chess[x1][y1] != "#" and (x1, y1,
i) not in visited:
                t1 = tire+abs(int(chess[x][y])-int(chess[x1][y1]))
                heapq.heappush(heap, (t1, x1, y1))
                visited.add((x1, y1, i))
    print(min(step) if step else "NO")
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

源代码

```
import heapq
m,n,t=map(int,input().split())
dirs=[(1,0),(-1,0),(0,1),(0,-1)]
chess=[list(input().split()) for i in range(m)]
for _ in range(t):
    sx, sy, ex, ey = map(int, input().split())
    if chess[sx][sy] == "#" or chess[ex][ey] == "#":
        print("NO")
        continue
    visited, heap,step= set(), [], []
    heapq.heappush(heap, (0, sx, sy))
    visited.add((sx, sy, -1))
    while heap:
        tire, x, y = heapq.heappop(heap)
        if x == ex and y == ey:
            step.append(tire)
        for i in range(4):
            dx, dy = dirs[i]
            x1, y1 = dx+x, dy+y
            if 0 <= x1 < m and 0 <= y1 < n and chess[x1][y1] != "#" and
                t1 = tire+abs(int(chess[x][y])-int(chess[x1][y1]))
                heapq.heappush(heap, (t1, x1, y1))
                visited.add((x1, y1, i))
    print(min(step) if step else "NO")
```

基本信息

#:	44837625
题目:	20106
提交人:	wangyecheng
内存:	4696kB
时间:	1659ms
语言:	Python3
提交时间:	2024-04-30 20:09:06

05442: 兔子与星空

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

思路:

代码

```
#
class DisjSet:
    def __init__(self, n):
        self.parent = [i for i in range(n)]
        self.rank = [0]*n

    def find(self, x):
        if self.parent[x] != x:
            self.parent[x] = self.find(self.parent[x])
        return self.parent[x]

    def union(self, x, y):
        xset, yset = self.find(x), self.find(y)
        if self.rank[xset] > self.rank[yset]:
            self.parent[yset] = xset
        else:
            self.parent[xset] = yset
            if self.rank[xset] == self.rank[yset]:
                self.rank[yset] += 1

def kruskal(n, edges):
    dset = DisjSet(n)
    edges.sort(key = lambda x:x[2])
    sol = 0
    for u, v, w in edges:
        u, v = ord(u)-65, ord(v)-65
        if dset.find(u) != dset.find(v):
            dset.union(u, v)
            sol += w
    if len(set(dset.find(i) for i in range(n))) > 1:
        return -1
    return sol

n = int(input())
edges = []
for _ in range(n-1):
    arr = input().split()
    root, m = arr[0], int(arr[1])
    for i in range(m):
        edges.append((root, arr[2+2*i], int(arr[3+2*i])))
print(kruskal(n, edges))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

基本信息

#: 44837651
题目: 05442
提交人: wangyecheng
内存: 3716kB
时间: 21ms
语言: Python3
提交时间: 2024-04-30 20:12:20

原代码

```
class DisjSet:
    def __init__(self, n):
        self.parent = [i for i in range(n)]
        self.rank = [0]*n

    def find(self, x):
        if self.parent[x] != x:
            self.parent[x] = self.find(self.parent[x])
        return self.parent[x]

    def union(self, x, y):
        xset, yset = self.find(x), self.find(y)
        if self.rank[xset] > self.rank[yset]:
            self.parent[yset] = xset
        else:
            self.parent[xset] = yset
            if self.rank[xset] == self.rank[yset]:
                self.rank[yset] += 1

def kruskal(n, edges):
    dset = DisjSet(n)
    edges.sort(key = lambda x:x[2])
    sol = 0
    for u, v, w in edges:
        u, v = ord(u)-65, ord(v)-65
        if dset.find(u) != dset.find(v):
            dset.union(u, v)
            sol += w
    if len(set(dset.find(i) for i in range(n))) > 1:
        return -1
    return sol

n = int(input())
edges = []
for _ in range(n-1):
    arr = input().split()
    root, m = arr[0], int(arr[1])
    for i in range(m):
        edges.append((root, arr[2+2*i], int(arr[3+2*i])))
print(kruskal(n, edges))
```

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。==

感觉慢慢对bfs和dfs熟悉了些，鸣人和走山路都是很好的练习题目，最后一题兔子与星空卡壳了，没有想到用并查集解决，看了题解后豁然开朗，继续加油！