

Assignment #A: 图论：算法，树算及栈

Updated 2018 GMT+8 Apr 21, 2024

2024 spring, Compiled by 赵策 数学科学学院

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：Visual Studio Code 1.86.2

1. 题目

20743: 整人的提词本

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

思路：

代码

```
#
s=input()
stack=[]
for i in s:
    if i==')':
        s1=''
        while stack[-1]!='(':
            s1+=stack.pop()
        stack.pop()
        stack.extend(s1)
    else:
        stack.append(i)
print(''.join(stack))
```

代码运行截图 (至少包含有"Accepted")

02255: 重建二叉树<http://cs101.openjudge.cn/practice/02255/>

思路:

代码

```
#
def build_tree(preorder, inorder):
    if not preorder or not inorder:
        return ''

    root=preorder[0]
    root_index=inorder.index(root)

    left_preorder=preorder[1:root_index+1]
    right_preorder=preorder[root_index+1:]

    left_inorder=inorder[:root_index]
    right_inorder=inorder[root_index+1:]

    left_tree=build_tree(left_preorder, left_inorder)
    right_tree=build_tree(right_preorder, right_inorder)

    return left_tree+right_tree+root

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

代码运行截图 (至少包含有"Accepted")

02255: 重建二叉树

Accepted

01426: Find The Multiple

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

要求用bfs实现

思路:

代码

```
#
from collections import deque
def find_multiple(n):
    queue=deque([1])
    while queue:
        m=queue.popleft()
        if m%n==0:
            return m
        m0=m*10
        m1=m*10+1
        queue.append(m0)
        queue.append(m1)

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

代码运行截图 (AC代码截图, 至少包含有"Accepted")

01426: Find The Multiple

Accepted

04115: 鸣人和佐助

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

思路:

代码

```
#
from collections import deque
def bfs():
    directions=[(0,1),(0,-1),(1,0),(-1,0)]
    while queue:
```

```

x,y,chakra,time=queue.popleft()
time+=1
for dx,dy in directions:
    nx,ny=x+dx,y+dy
    if 0<=nx<m and 0<=ny<n:
        elem=grid[nx][ny]
        if elem=='*' and chakra>visited[nx][ny]:
            visited[nx][ny]=chakra
            queue.append((nx,ny,chakra,time))
        elif elem=='#' and chakra-1>visited[nx][ny]:
            visited[nx][ny]=chakra-1
            queue.append((nx,ny,chakra-1,time))
        elif elem=='+':
            return time
    return -1

m,n,t=map(int,input().split())
grid=[input() for _ in range(m)]
start=None
visited=[[-1]*n for _ in range(m)]
for i in range(m):
    for j in range(n):
        if grid[i][j]=='@':
            start=(i,j)
            visited[i][j]=t
if start:
    break
queue=deque([start+(t,0)])
print(bfs())

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

04115: 鸣人和佐助

Accepted

20106: 走山路

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

思路:

代码

```

#
import heapq
def bfs():
    if grid[x1][y1]=='#' or grid[x2][y2]=='#':
        return 'NO'
    direc=[(-1,0),(1,0),(0,1),(0,-1)]

```

```

d=[[float('inf')]*n for _ in range(m)]
d[x1][y1]=0
points=[(0,x1,y1)]
while points:
    stamina,x,y=heapq.heappop(points)
    h=grid[x][y]
    if (x,y)==(x2,y2):
        return stamina
    for dx,dy in direc:
        nx,ny=x+dx,y+dy
        if 0<=nx<m and 0<=ny<n and grid[nx][ny]!='#' and d[nx]
[ny]>stamina+abs(grid[nx][ny]-h):
            d[nx][ny]=stamina+abs(grid[nx][ny]-h)
            heapq.heappush(points,(d[nx][ny],nx,ny))
    return 'NO'

m,n,p=map(int,input().split())
grid=[[int(x) if x!='#' else '#' for x in input().split()] for _ in range(m)]
for _ in range(p):
    x1,y1,x2,y2=map(int,input().split())
    print(bfs())

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

20106: 走山路

Accepted

05442: 兔子与星空

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

思路:

代码

```
#
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

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。

dfs,bfs变体太多了，这次作业耗时还是有点长