

Assignment #F: All-Killed 满分

Updated 1844 GMT+8 May 20, 2024

2024 spring, Compiled by 尹柚鑫 2100015878 光华管理学院

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: Win11

Python编程环境: jupyter notebook

1. 题目

22485: 升空的焰火，从侧面看

<http://cs101.openjudge.cn/practice/22485/>

耗时: 15mins

思路: 使用BFS遍历每一层，保留每一层的最后一个即可，助教这个用for循环自动保存了最后一个的写法很巧妙，比我自己写的简洁多了

代码

```
#
from collections import deque

def right_view(n, tree):
    queue = deque([(1, tree[1])]) # start with root node
    right_view = []

    while queue:
        level_size = len(queue)
        for i in range(level_size):
            node, children = queue.popleft()
            if children[0] != -1:
```

```

        queue.append((children[0], tree[children[0]]))
        if children[1] != -1:
            queue.append((children[1], tree[children[1]]))
        right_view.append(node)

    return right_view

n = int(input())
tree = {1: [-1, -1]} # initialize tree with -1s
for i in range(1, n+1):
    left, right = map(int, input().split())
    tree[i] = [left, right]

result = right_view(n, tree)
print(' '.join(map(str, result)))

```

代码运行截图 == (至少包含有"Accepted") ==

#45122136提交状态

统计 提问

状态: **Accepted**

源代码

```

from collections import deque

def right_view(n, tree):
    queue = deque([(1, tree[1])]) # start with root node
    right_view = []

    while queue:
        level_size = len(queue)

```

基本信息

#: 45122136
 题目: 22485
 提交人: 尹柚鑫(2100015878)
 内存: 3740kB
 时间: 22ms
 语言: Python3
 提交时间: 2024-05-28 20:59:05

28203: 【模板】单调栈

<http://cs101.openjudge.cn/practice/28203/>

耗时: 10mins

思路: 单调栈模版题目, 感觉和动态规划的想法有一点像, 一些单调栈的题目也可以用动态规划写

代码

```

#
n=int(input())
numlist=[*map(int,input().split())]
b=[0]*len(numlist)
stack=[]
for i in range(n):
    while stack and numlist[stack[-1]]<numlist[i]:
        b[stack.pop()]=i+1
    stack.append(i)
while stack:
    numlist[stack[-1]]=0
    stack.pop()
print(*b)

```

代码运行截图 == (至少包含有"Accepted") ==

#45122509提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
n=int(input())
numlist=[*map(int,input().split())]
b=[0]*len(numlist)
stack=[]
for i in range(n):
    while stack and numlist[stack[-1]]<numlist[i]:
        b[stack.pop()]=i+1
    stack.append(i)
```

基本信息

#: 45122509
题目: 28203
提交人: 尹柚鑫(2100015878)
内存: 384816kB
时间: 3001ms
语言: Python3
提交时间: 2024-05-28 21:22:07

09202: 舰队、海域出击!

<http://cs101.openjudge.cn/practice/09202/>

思路：一开始和之前的无向图判断是否成环有一点点像，但是一直没有过。参考了答案的写法，感觉是比较规整的算法，要记一记

代码

```
#
from collections import defaultdict

def dfs(node, color):
    color[node] = 1
    for neighbour in graph[node]:
        if color[neighbour] == 1:
            return True
        if color[neighbour] == 0 and dfs(neighbour, color):
            return True
    color[node] = 2
    return False

T = int(input())
for _ in range(T):
    N, M = map(int, input().split())
    graph = defaultdict(list)
    for _ in range(M):
        x, y = map(int, input().split())
        graph[x].append(y)
    color = [0] * (N + 1)
    is_cyclic = False
    for node in range(1, N + 1):
        if color[node] == 0:
            if dfs(node, color):
                is_cyclic = True
                break
    print("Yes" if is_cyclic else "No")
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

#45122890提交状态

状态: Accepted

源代码

```
from collections import defaultdict

def dfs(node, color):
    color[node] = 1
    for neighbour in graph[node]:
        if color[neighbour] == 1:
            return True
        if color[neighbour] == 0 and dfs(neighbour, color):
            return True
    color[node] = 2
    return False
```

04135: 月度开销

<http://cs101.openjudge.cn/practice/04135/>

思路: 自己写感觉一头雾水, 我的想法是, 先靠左一个一个放板子, 再调整, 但是发现给不出来调整策略。看了一下题解, 今天太晚了, 刚写完一个数学的项目, 脑子实在转不动了, 没有看懂, 明天起来再仔细研究

代码

```
#
n,m = map(int, input().split())
expenditure = []
for _ in range(n):
    expenditure.append(int(input()))

def check(x):
    num, s = 1, 0
    for i in range(n):
        if s + expenditure[i] > x:
            s = expenditure[i]
            num += 1
        else:
            s += expenditure[i]

    return [False, True][num > m]
```

```
# https://github.com/python/cpython/blob/main/Lib/bisect.py
lo = max(expenditure)
# hi = sum(expenditure)
hi = sum(expenditure) + 1
ans = 1
while lo < hi:
    mid = (lo + hi) // 2
    if check(mid):      # 返回True, 是因为num>m, 是确定不合适
        lo = mid + 1    # 所以lo可以置为 mid + 1。
    else:
        ans = mid       # 如果num==m, mid可能是答案
        hi = mid

#print(lo)
print(ans)
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

#45123193提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
n,m = map(int, input().split())
expenditure = []
for _ in range(n):
    expenditure.append(int(input()))

def check(x):
    num, s = 1, 0
```

基本信息

#: 45123193
 题目: 04135
 提交人: 尹柚鑫(2100015878)
 内存: 7484kB
 时间: 555ms
 语言: Python3
 提交时间: 2024-05-28 22:09:54

07735: 道路

<http://cs101.openjudge.cn/practice/07735/>

思路:

代码

```
#
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

01182: 食物链

<http://cs101.openjudge.cn/practice/01182/>

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。==

期末临近, 挺多以课程项目结课的课褶皱占有了我的大量时间, 这一周要疯狂总结回顾和做题巩固了

一学期的数算课程, 过得很快, 有难度, 学到了很多东西。同时数算也培养了我一种新的思想, 程序强大的计算能力让很多解题过程和数学很不一样, 让我感觉很奇妙

明年我就大四了, 感觉要趁毕业前再多学一点计算机的课程, 非常有用, 并且有意思