

Assignment #F: All-Killed 满分

Updated 1844 GMT+8 May 20, 2024

2024 spring, Compiled by 同学的姓名、院系

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

22485: 升空的焰火，从侧面看

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

思路:

送分bfs

代码

```
N=int(input())
childdct={}
for i in range(N):
    l,r=map(int,input().split())
    childdct[i+1]=(l,r)
def bfs():
    currentlevel=[1]
    nextlevel=[]
    ans=[1]
    while True:
        for node in currentlevel:
```

```
l,r=childdct[node]
if l!=-1:
    nextlevel.append(l)
if r!=-1:
    nextlevel.append(r)
if not nextlevel:
    return ans
ans.append(nextlevel[-1])
currentlevel=nextlevel
nextlevel=[]

for x in bfs():
    print(x,end=' ')
```

代码运行截图 (至少包含有"Accepted")

OpenJudge 题目ID, 标题, 描述

 **CS101 / 题库**

题目 排名 状态 提问

#45028019提交状态

查看 损

状态: **Accepted**

源代码

```
N=int(input())
childdct={}
for i in range(N):
    l,r=map(int,input().split())
    childdct[i+1]=(l,r)
def bfs():
    currentlevel=[1]
    nextlevel=[]
    ans=[1]
    while True:
        for node in currentlevel:
            l,r=childdct[node]
            if l!=-1:
                nextlevel.append(l)
            if r!=-1:
                nextlevel.append(r)
        if not nextlevel:
            return ans
    ans.append(nextlevel[-1])
    currentlevel=nextlevel
    nextlevel=[]

for x in bfs():
    print(x,end=' ')
```

基本信息

#: 4502801

题目: 22485

提交人: zxk

内存: 3672kB

时间: 21ms

语言: Python3

提交时间: 2024-05

28203:【模板】单调栈

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

思路:

???

代码

```
n=int(input())
data=[int(x) for x in input().split()]
def f(i):
    for j in range(i+1,n):
        if data[j]>data[i]:
            return j+1
    return 0
for i in range(n):
    print(f(i),end=' ')
```

代码运行截图 (至少包含有"Accepted")

#45028123提交状态

状态: Accepted

源代码

```
n=int(input())
data=[int(x) for x in input().split()]
def f(i):
    for j in range(i+1,n):
        if data[j]>data[i]:
            return j+1
    return 0
for i in range(n):
    print(f(i),end=' ')
```

总时间限制: 20000ms

基本信息

#: 45028123

题目: 28203

提交人: zxk

内存: 359944kB

时间: 6175ms

语言: Python3

提交时间: 2024-05-20 20:20:03

09202: 舰队、海域出击!

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

思路:

dfs很好想, 问题出在了时间限制上, 数据规模极大, 导致对于dfs的判断需要很迅速, 对于处理过的节点就不应该再进行dfs了

代码

```
def hasCycle(graph, v, visited, recStack):
    visited.add(v)
    recStack.add(v)
    for nei in graph[v]:
        if nei not in visited:
            if hasCycle(graph, nei, visited, recStack):
                return True
        elif nei in recStack:
            return True
    recStack.remove(v)
    return False

T = int(input())
anslst = []
for _ in range(T):
    N, M = map(int, input().split())
    graph = {i: set() for i in range(1, N+1)}
    for _ in range(M):
        x, y = map(int, input().split())
        graph[x].add(y)

    ans = 'No'
    visited = set()
    for i in range(1, N+1):
        if hasCycle(graph, i, visited, set()):
            ans = 'Yes'
            break
    anslst.append(ans)
```

```
for ans in anslst:
    print(ans)
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

OpenJudge

题目ID, 标题, 描述

zxk

信箱

CS101 / 题库

题目

排名

状态

提问

#45087594提交状态

查看

提交

统计

状态: Accepted

基本信息

源代码

```
def hasCycle(graph, v, visited, recStack):
    visited.add(v)
    recStack.add(v)
    for nei in graph[v]:
        if nei not in visited:
            if hasCycle(graph, nei, visited, recStack):
                return True
        elif nei in recStack:
            return True
    recStack.remove(v)
    return False

T = int(input())
anslst = []
for _ in range(T):
    N, M = map(int, input().split())
    graph = {i: set() for i in range(1, N+1)}
    for _ in range(M):
        x, y = map(int, input().split())
        graph[x].add(y)

    ans = 'No'
    visited = set()
    for i in range(1, N+1):
        if hasCycle(graph, i, visited, set()):
            ans = 'Yes'
            break
    anslst.append(ans)

for ans in anslst:
    print(ans)
```

#: 45087594

题目: 09202

提交人: zxk

内存: 109268kB

时间: 4710ms

语言: Python3

提交时间: 2024-05-25 21:39:3

©2002-2022 POJ 京ICP备20010980号-1

English

帮!

04135: 月度开销

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

思路:

代码

```

def check(x,m,dlst):
    month_cnt=1
    stack_weight=0
    for i in range(len(dlst)):
        p=dlst[i]
        if stack_weight+p>x:
            month_cnt+=1
            stack_weight=p
            if stack_weight>x:
                return False
        else:
            stack_weight+=p
    # print('deal with:',x,f'get {month_cnt}')
    return month_cnt<=m

N,M=map(int,input().split())
data=[]
for _ in range(N):
    data.append(int(input()))

l,r=max(data),sum(data)
m=None
while l<r:
    m=(l+r)//2
    if check(m,M,data):
        r=m
    else:
        l=m+1#这一步很关键，假设应该输出m，如果缩到m-1,m+1:m-1,m，均值m-1，缩到m,m，退出；如果缩到m,m+1:均值取m，能过，缩到m,m退出；
print(l)

```

代码运行截图 (AC代码截图，至少包含有"Accepted")



#45106780提交状态

[查看](#) [提交](#) [统计](#)

状态: Accepted

基本信息

#: 45106780

题目: 04135

提交人: zxk

内存: 7492kB

时间: 460ms

语言: Python3

提交时间: 2024-05-27 15:07:5

源代码

```
def check(x,m,dlst):
    month_cnt=1
    stack_weight=0
    for i in range(len(dlst)):
        p=dlst[i]
        if stack_weight+p>x:
            month_cnt+=1
            stack_weight=p
            if stack_weight>x:
                return False
        else:
            stack_weight+=p
    # print('deal with:',x,f'get {month_cnt}')
    return month_cnt<=m

N,M=map(int,input().split())
data=[]
for _ in range(N):
    data.append(int(input()))

l,r=max(data),sum(data)
m=None
while l<r:
    m=(l+r)//2
    if check(m,M,data):
        r=m
    else:
        l=m+1
print(l)
```

07735: 道路

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

思路:

带有cost的dijk, 开始的时候不会做, 总感觉不是dijk, 用bfs试了试, 应该是陷入死循环出不来了, 要么就是数据量太大; 问了一下gpt发现是dijk, 只不过是优先判断能不能走通; 感觉思路里最妙的地方在于dijk的动态表是二维的, 记录了节点和走到这里的花费, 这就避免了因为花费而影响判断

代码

```
import heapq

K=int(input())
N=int(input())
```

```

R=int(input())
g=[]for i in range(N)
for i in range(R):
    f,t,l,c=map(int,input().split())
    g[f-1].append([t-1,l,c])

que=[(0,0,0)]
dist=[[999999999]*(K+1) for _ in range(N)]
dist[0][0]=0

while que:
    current_l,current_node,current_cost=heapq.heappop(que)
    if current_l>dist[current_node][current_cost]:
        continue

    for new_ver,length,cost in g[current_node]:
        if current_cost+cost<=K and current_l+length<dist[new_ver]
[current_cost+cost]:
            dist[new_ver][current_cost+cost]=current_l+length
            heapq.heappush(que,(current_l+length,new_ver,current_cost+cost))

min_dist=min(dist[N-1])
if min_dist!=999999999:
    print(min_dist)
else:
    print(-1)

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

CS101 / 题库 (包括计概、数算题目)

题目排名状态提问

#45116732提交状态

查看提交统计

状态: Accepted

源代码

```
import heapq

K=int(input())
N=int(input())
R=int(input())
g=[[[]for i in range(N)]
for i in range(R):
    f,t,l,c=map(int,input().split())
    g[f-1].append([t-1,l,c])

que=[(0,0,0)]
dist=[[9999999999]*(K+1) for _ in range(N)]
dist[0][0]=0

while que:
    current_l,current_node,current_cost=heapq.heappop(que)
    if current_l>dist[current_node][current_cost]:
        continue

    for new_ver,length,cost in g[current_node]:
        if current_cost+cost<=K and current_l+length<dist[new_ver][current_cost+cost]:
            dist[new_ver][current_cost+cost]=current_l+length
            heapq.heappush(que,(current_l+length,new_ver,current_cost+cost))

min_dist=min(dist[N-1])
if min_dist!=9999999999:
    print(min_dist)
else:
    print(-1)
```

基本信息

#: 45116732

题目: 07735

提交人: zxx

内存: 30644kB

时间: 3266ms

语言: Python3

提交时间: 2024-05-28 15:04:0

©2002-2022 POJ 京ICP备20010980号-1

English帮助

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。

·焰火简单

·单调栈为啥给了20s的限制时间啊？

·舰队：有向图判断环路，学了一下拓扑排序，但是感觉很难实现，用dfs也会莫名其妙的出错；可以把这个代码cheat下来

·月度开销：折磨人...思路只要知道了二分查找就很简单，但是这个循环的语法还挺乱的，试了好多遍，总是有问题，最终还是借鉴了题解，感觉这种循环真的很折磨人；对于不同的逼近方式，退出的条件不一样很难保证对所有数据都合适；

·道路：带有cost的dijk，之前鸣人佐助就没写明白，这次基本明白了，但是还是需要记下来

·食物链：看了一下午也没想出来怎么实现，思路看了个带权值路径的并查集，通过权值求和模3确定两个节点的关系，有了这个关系应该就和普通并查集一样了，先交上作业，代码之后再补

这次作业折磨了一个星期，各种不会，看来图对于我来说还是有点太难了 😭 希望下周机考可以做到AC4，不求多高的分数，只求稳健一点就好了 😊