

Assignment #D: May月考

Updated 1654 GMT+8 May 8, 2024

2024 spring, Compiled by 夏天明 元培学院

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 10 | 22H2

Python编程环境: Spyder IDE 5.4.3 | Python 3.11.4 64-bit

1. 题目

02808: 校门外的树

<http://cs101.openjudge.cn/practice/02808/>

思路: 直接模拟

代码

```
L, M = map(int, input().split())
tree = [1]*(L+1)
for o in range(M):
    start, end = map(int, input().split())
    for i in range(start, end+1):
        tree[i] = 0
print(sum(tree))
```

代码运行截图

状态: Accepted

源代码

```
L, M = map(int, input().split())
tree = [1]*(L+1)
for o in range(M):
    start, end = map(int, input().split())
    for i in range(start, end+1):
        tree[i] = 0
print(sum(tree))
```

基本信息

#: 44897383

题目: E02808

提交人: 23n2300017735(夏天明)

BrightSummer)

内存: 3636kB

时间: 41ms

语言: Python3

提交时间: 2024-05-08 15:13:15

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

20449: 是否被5整除

<http://cs101.openjudge.cn/practice/20449/>

思路:

```
class int(x=0)
```

```
class int(x, base=10)
```

返回一个基于数字或字符串 x 构造的整数对象，或者在未给出参数时返回 0。如果 x 定义了 `__int__()`，则 `int(x)` 将返回 `x.__int__()`。如果 x 定义了 `__index__()`，则将返回 `x.__index__()`。如果 x 定义了 `__trunc__()`，则将返回 `x.__trunc__()`。对于浮点数，这将向零方向截断。

如果 x 不是一个数字或者如果给定了 $base$ ，则 x 必须是一个表示以 $base$ 为基数的整数的字符串、`bytes` 或 `bytearray` 实例。字符串前面还能加上可选的 `+` 或 `-` (中间没有空格)，带有前导的零，带有两侧的空格，并可带有数位之间的单个下划线。

一个以 n 为基数的整数字符串包含多个数位，每个数位代表从 0 到 $n-1$ 范围内的值。0--9 的值可以用任何 Unicode 十进制数码来表示。10--35 的值可以用 `a` 到 `z` (或 `A` 到 `Z`) 来表示。默认的 $base$ 为 10。允许的基数为 0 和 2--36。对于基数 2, -8 和 -16 来说字符串前面还能加上可选的 `0b/0B`, `0o/0O` 或 `0x/0X` 前缀，就像代码中的整数字面值那样。对于基数 0 来说，字符串会以与 [代码中的整数字面值](#) 类似的方式来解读，即实际的基数将由前缀确定为 2, 8, 10 或 16。基数为 0 还会禁用前导的零: `int('010', 0)` 将是无效的，而 `int('010')` 和 `int('010', 8)` 则是有效的。

代码

```
s = input()
print(''.join('1' if int(s[:i+1], 2)%5 == 0 else '0' for i in range(len(s))))
```

代码运行截图

状态: Accepted

源代码

```
s = input()
print(''.join('1' if int(s[:i+1], 2)%5 == 0 else '0' for i in range(len(s)
```

基本信息

#: 44960263

题目: 20449

提交人: 23n2300017735(夏天
BrightSummer)

内存: 3596kB

时间: 21ms

语言: Python3

01258: Agri-Net

<http://cs101.openjudge.cn/practice/01258/>

思路: Prim模板题

代码

```
from heapq import heappop, heappush

while True:
    try:
        N = int(input())
    except EOFError:
        break
    distance = [[int(i) for i in input().split()] for j in range(N)]
    cost = [float('inf')]*N
    q = [(0, 0)]
    solved = [False]*N
    ans = 0
    cost[0] = 0
    while q:
        c, a = heappop(q)
        if not solved[a]:
            ans += c
            solved[a] = True
            for i in range(N):
                if not solved[i]:
                    cost[i] = min(cost[i], distance[a][i])
                    heappush(q, (cost[i], i))
    print(ans)
```

代码运行截图

状态: Accepted

源代码

```
from heapq import heappop, heappush

while True:
    try:
        N = int(input())
    except EOFError:
        break
    distance = [[int(i) for i in input().split()] for j in range(N)]
    cost = [float('inf')] * N
    q = [(0, 0)]
    solved = [False] * N
    ans = 0
    cost[0] = 0
    while q:
        c, a = heappop(q)
        if not solved[a]:
            ans += c
            solved[a] = True
            for i in range(N):
                if not solved[i]:
                    cost[i] = min(cost[i], distance[a][i])
                    heappush(q, (cost[i], i))
    print(ans)
```

基本信息

#: 44897667
题目: M01258
提交人: 23n2300017735(夏天明
BrightSummer)
内存: 4288kB
时间: 43ms
语言: Python3
提交时间: 2024-05-08 15:28:39

27635: 判断无向图是否连通有无回路(同23163)

<http://cs101.openjudge.cn/practice/27635/>

思路: 直接用dfs解决

代码

```
def dfs(p):
    visited[p] = 1
    for q in nex[p]:
        if not visited[q]:
            dfs(q)

def isLoop(p, parent):
    visited2[p] = 1
    for q in nex[p]:
        if not visited2[q]:
            if isLoop(q, p):
                return True
        else:
            if q != parent:
                return True

n, m = map(int, input().split())
nex = [[] for i in range(n)]
for o in range(m):
```

```

u, v = map(int, input().split())
nex[u].append(v)
nex[v].append(u)
visited = [0]*n
dfs(0)
print(f"connected: {'yes' if all(visited) else 'no'}")
visited2 = [0]*n
loop = False
for i in range(n):
    if not visited2[i] and isLoop(i, -1):
        loop = True
        break
print(f"loop: {'yes' if loop else 'no'}")

```

代码运行截图

#44898039提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

def dfs(p):
    visited[p] = 1
    for q in nex[p]:
        if not visited[q]:
            dfs(q)

def isLoop(p, parent):
    visited2[p] = 1
    for q in nex[p]:
        if not visited2[q]:
            if isLoop(q, p):
                return True
        else:
            if q != parent:
                return True

n, m = map(int, input().split())
nex = [[] for i in range(n)]
for o in range(m):
    u, v = map(int, input().split())
    nex[u].append(v)
    nex[v].append(u)
visited = [0]*n
dfs(0)

```

基本信息

#: 44898039
 题目: M27635
 提交人: 23n2300017735(夏天明
 BrightSummer)
 内存: 3720kB
 时间: 26ms
 语言: Python3
 提交时间: 2024-05-08 15:41:32

27947: 动态中位数

<http://cs101.openjudge.cn/practice/27947/>

思路: 维护两个堆, 并且保持左堆的元素都比右堆小, 且元素个数不大于右堆元素个数+1, 那么当压入元素的总个数是奇数时, 左堆堆顶就是中位数。

代码

```

from heapq import heappush, heappop

for o in range(int(input())):
    arr = [int(i) for i in input().split()]
    L, R = [-arr[0]], []
    ans = [arr[0]]
    for i in range(1, len(arr)):
        if arr[i] < -L[0]:
            heappush(L, -arr[i])
        else:
            heappush(R, arr[i])
        if len(L) > len(R)+1:
            heappush(R, -heappop(L))
        elif len(R) > len(L):
            heappush(L, -heappop(R))
        if not i&1:
            ans.append(-L[0])
    print(len(ans))
    print(*ans)

```

代码运行截图

#44898379提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

from heapq import heappush, heappop

for o in range(int(input())):
    arr = [int(i) for i in input().split()]
    L, R = [-arr[0]], []
    ans = [arr[0]]
    for i in range(1, len(arr)):
        if arr[i] < -L[0]:
            heappush(L, -arr[i])
        else:
            heappush(R, arr[i])
        if len(L) > len(R)+1:
            heappush(R, -heappop(L))
        elif len(R) > len(L):
            heappush(L, -heappop(R))
        if not i&1:
            ans.append(-L[0])
    print(len(ans))
    print(*ans)

```

基本信息

#: 44898379
 题目: T27947
 提交人: 23n2300017735(夏天明
 BrightSummer)
 内存: 10656kB
 时间: 350ms
 语言: Python3
 提交时间: 2024-05-08 15:55:06

28190: 奶牛排队

<http://cs101.openjudge.cn/practice/28190/>

思路：遍历所有点作为右端点 r ，则左端点 l 满足： l 是 $[0, r]$ 的后缀**严格**最小值点，且 $[l, r]$ 不含除 r 外的后缀**非严格**最大值点，即 l 位于 $[0, r]$ 的倒数第二个后缀非严格最大值点的严格右面。

因此关键在于维护 $[0, r]$ 的后缀严格最小值点列 low 和后缀非严格最大值点列 $high$ 。使用单调栈完成。那么 l 的最优解是 $high[-2]$ 严格右面的第一个 low 中的值，由于 low 里面的点是从左到右入栈的，所以使用`bisect_right`搜索即可。

代码

```
from bisect import bisect_right

h = []
low, high = [], []
ans = 0
for i in range(int(input())):
    h.append(int(input()))
    while low and h[-1] <= h[low[-1]]:
        low.pop()
    low.append(i)
    while high and h[-1] > h[high[-1]]:
        high.pop()
    high.append(i)
    ans = max(ans, i - low[bisect_right(low, high[-2] if len(high)>1 else -1)])
print(ans+1 if ans else 0)
```

代码运行截图

#44968819提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
from bisect import bisect_right

h = []
low, high = [], []
ans = 0
for i in range(int(input())):
    h.append(int(input()))
    while low and h[-1] <= h[low[-1]]:
        low.pop()
    low.append(i)
    while high and h[-1] > h[high[-1]]:
        high.pop()
    high.append(i)
    ans = max(ans, i - low[bisect_right(low, high[-2] if len(high)>1 else -1)])
print(ans+1 if ans else 0)
```

基本信息

#: 44968819
题目: 28190
提交人: 23n2300017735(夏天明
BrightSummer)
内存: 47780kB
时间: 2383ms
语言: Python3
提交时间: 2024-05-15 13:45:50

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

2. 学习总结和收获

本次月考深入浅出，前面几道题是基础知识的复习，最后两道题较为新颖，乍一看上去不是很好做，好在有tag的提示，提供了思考的切入点。即便如此，这两道题同样涉及很多细节，要写对还是要花很大功夫。