Assignment #F: All-Killed 满分

Updated 1844 GMT+8 May 20, 2024

2024 spring, Complied by 数学科学学院 王镜廷 2300010724

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：Windows11 专业版

Python编程环境：VSCode 1.86.2, with extension Python and python version 3.12.2

1. 题目

22485: 升空的焰火，从侧面看

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

用时：约15分钟

思路：建树后BFS的同时记录深度，如果弹出一个元素之后队列为空或队列中下一元素深度大于该元素，则说明该元素是同一层最右边的节点，加入答案中。

代码

```python
from collections import deque

n = int(input())
left = [0]
right = [0]
fa = [0] * (n + 1)
h = {}
for i in range(n):
    u, v = map(int, input().split())
    left.append(u)
    right.append(v)
    if u != -1:
        fa[u] = i + 1
    if v != -1:
        fa[v] = i + 1
root = 0
for i in range(n):
    if fa[i + 1] == 0:
        root = i + 1
q = deque()
q.append(root)
h[root] = 0
res = []
# print(left, right)
while q:
    u = q.popleft()
    # print(u, q, h)
    if left[u] != -1:
        q.append(left[u])
        h[left[u]] = h[u] + 1
    if right[u] != -1:
        q.append(right[u])
        h[right[u]] = h[u] + 1
    if not q:
        res.append(u)
        continue
    v = q.popleft()
    if h[v] > h[u]:
        res.append(u)
    q.appendleft(v)
print(" ".join(str(i) for i in res))
```

代码运行截图 （至少包含有"Accepted"）

#45064938提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque

n = int(input())
left = [0]
right = [0]
fa = [0] * (n + 1)
h = {}
for i in range(n):
    u, v = map(int, input().split())
    left.append(u)
    right.append(v)
    if u != -1:
        fa[u] = i + 1
    if v != -1:
        fa[v] = i + 1
root = 0
for i in range(n):
    if fa[i + 1] == 0:
        root = i + 1
```

基本信息

#: 45064938
题目: 22485
提交人: 23n2300010724
内存: 3772kB
时间: 23ms
语言: Python3
提交时间: 2024-05-24 15:00:13

28203:【模板】单调栈

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

用时：约15分钟

思路：直接写模板即可，一开始MLE了，看了群里的消息发现是输出方式的问题

代码

```
n = int(input())
s = list(map(int, input().split()))
q = []
res = []
for i in range(n - 1, -1, -1):
    while q and s[i] >= s[q[-1]]:
        q.pop()
    res.append(-1 if not q else q[-1])
    q.append(i)
res.reverse()
for i in res:
    print(i + 1, end=" ")
```

代码运行截图 （至少包含有"Accepted"）

#45065264提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n = int(input())
s = list(map(int, input().split()))
q = []
res = []
for i in range(n - 1, -1, -1):
    while q and s[i] >= s[q[-1]]:
        q.pop()
    res.append(-1 if not q else q[-1])
    q.append(i)
res.reverse()
for i in res:
    print(i + 1, end=" ")
```

基本信息

#: 45065264
题目: 28203
提交人: 23n2300010724
内存: 384132kB
时间: 3808ms
语言: Python3
提交时间: 2024-05-24 15:15:49

English 帮助 关于

09202: 舰队、海域出击!

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

用时：约9分钟

思路：用拓扑排序的类似方法判断有向图是否有环。

代码

```python
T = int(input())
for case in range(T) :
    n, m = map(int, input().split())
    degree = [0]
    edge = [[]]
    for i in range(1, n + 1):
        edge.append([])
        degree.append(0)
    for i in range(m):
        u, v = map(int, input().split())
        edge[u].append(v)
        degree[v] += 1
    cnt = 0
    q = []
    for i in range(1, n + 1):
        if degree[i] == 0:
            q.append(i)
    while q:
        u = q[-1]
        cnt += 1
        q.pop()
        for v in edge[u]:
            degree[v] -= 1
            if degree[v] == 0:
                q.append(v)
    print("No" if cnt == n else "Yes")
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#45065362提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 45065362
题目: 09202
提交人: 23n2300010724
内存: 40096kB
时间: 3704ms
语言: Python3
提交时间: 2024-05-24 15:24:00

源代码

```
T = int(input())
for case in range(T) :
    n, m = map(int, input().split())
    degree = [0]
    edge = [[]]
    for i in range(1, n + 1):
        edge.append([])
        degree.append(0)
    for i in range(m):
        u, v = map(int, input().split())
        edge[u].append(v)
        degree[v] += 1
    cnt = 0
    q = []
    for i in range(1, n + 1):
        if degree[i] == 0:
            q.append(i)
    while q:
        u = q[-1]
        cnt += 1
        q.pop()
        for v in edge[u]:
```

04135: 月度开销

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

用时：约7分钟

思路：先二分，从而只需判断每个周期花销不超过m时所分得的周期数是否不超过M，而每个周期最大花销固定时最小周期数可以贪心求得。

代码

```python
n, m = map(int, input().split())
s = []
for i in range(n) :
    s.append(int(input()))
def check(M) :
    if M < max(s) :
        return False
    cnt = 1
    tmp = 0
    for i in range(n):
        if tmp + s[i] > M:
            cnt += 1
            tmp = s[i]
        else :
            tmp += s[i]
    return cnt <= m

L, R = 0, sum(s)
while R > L :
    M = (L + R) // 2
    if not check(M):
        L = M + 1
    else :
        R = M
print(L)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#45065456提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```python
n, m = map(int, input().split())
s = []
for i in range(n) :
    s.append(int(input()))
def check(M) :
    if M < max(s) :
        return False
    cnt = 1
    tmp = 0
    for i in range(n):
        if tmp + s[i] > M:
            cnt += 1
            tmp = s[i]
        else :
            tmp += s[i]
    return cnt <= m

L, R = 0, sum(s)
while R > L :
    M = (L + R) // 2
    if not check(M):
        L = M + 1
    else :
        R = M
```

基本信息

#: 45065456
题目: 04135
提交人: 23n2300010724
内存: 7928kB
时间: 527ms
语言: Python3
提交时间: 2024-05-24 15:31:03

07735: 道路

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

用时：约25分钟

思路：类似于分层图最短路，考虑点为(u, c)即原图中走到点u，花费金币数c时的最短路程。一开始MLE了，后来看了看题解发现应该在第一次搜到形如(n, c)的点时直接返回这是最短路即可。（这大概说明数据里面没有n - 1个点之间很连通，第n个点完全不跟别的点相连这样的例子，或者这里的最短路径受到金币限制而很长之类的例子，可能应该考虑增强数据，但是增强之后我也没有很好的剪枝方法了）

代码

```python
import heapq

K = int(input())
N = int(input())
M = int(input())
edge = [[]]
for i in range(N):
    edge.append([])
for i in range(M) :
    u, v, w, c = map(int, input().split())
    edge[u].append((v, w, c))
dist = {}
q = [(0, 1, 0)]
vis = set()
heapq.heapify(q)
while q:
    d, u, c = heapq.heappop(q)
    if (u, c) in vis:
        continue
    dist[(u, c)] = d
    vis.add((u, c))
    if u == N :
        break
    for e in edge[u]:
        v, w, c1 = e
        w2, c2 = d + w, c1 + c
        if c2 > K:
            continue
        if (v, c2) not in dist or dist[(v, c2)] > w2:
            heapq.heappush(q, (w2, v, c2))
res = -1
for i in dist:
    u = i[0]
    if u == N:
        res = dist[i] if res == -1 else min(res, dist[i])
print(res)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#45065787提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
import heapq

K = int(input())
N = int(input())
M = int(input())
edge = [[]]
for i in range(N):
    edge.append([])
for i in range(M) :
    u, v, w, c = map(int, input().split())
    edge[u].append((v, w, c))
dist = {}
q = [(0, 1, 0)]
vis = set()
heapq.heapify(q)
while q:
    d, u, c = heapq.heappop(q)
    if (u, c) in vis:
```

基本信息

#: 45065787
题目: 07735
提交人: 23n2300010724
内存: 6604kB
时间: 51ms
语言: Python3
提交时间: 2024-05-24 15:55:08

01182: 食物链

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

用时：约7分钟

思路：视作一个简化版的2-SAT问题，并查集中节点对应于一点u在A，u在B，u在C，因为条件都可以视为等价关系，从而维护并查集实现。

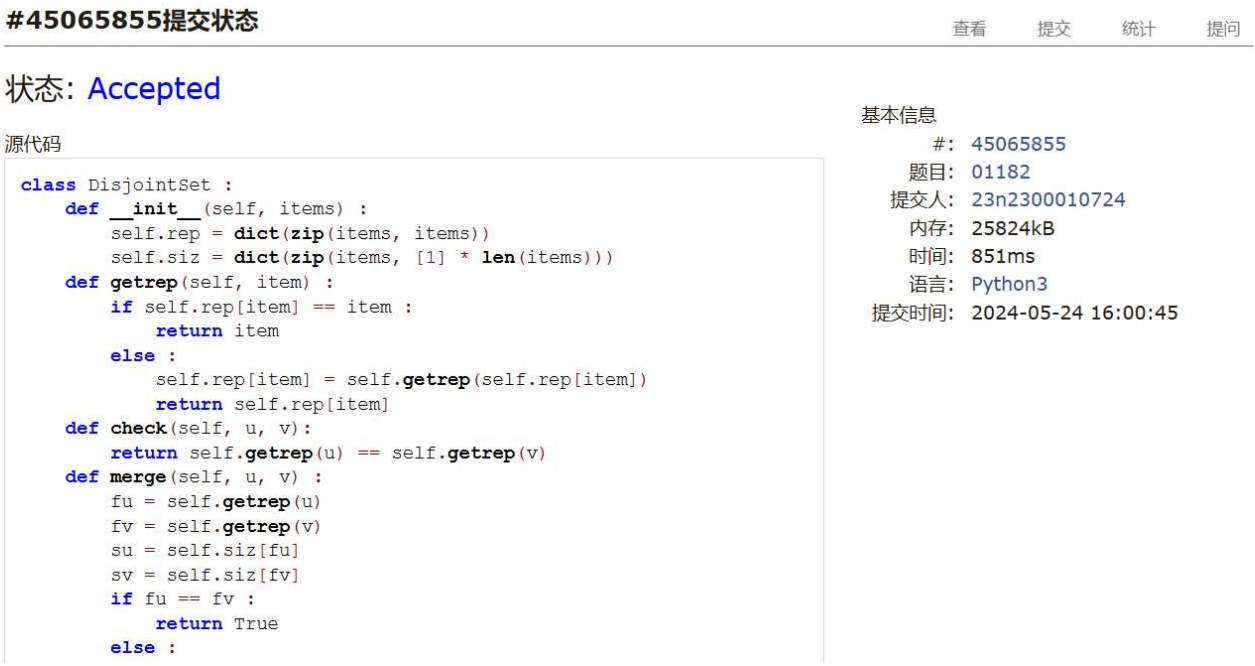
代码

```python
class DisjointSet :
    def __init__(self, items) :
        self.rep = dict(zip(items, items))
        self.siz = dict(zip(items, [1] * len(items)))
    def getrep(self, item) :
        if self.rep[item] == item :
            return item
        else :
            self.rep[item] = self.getrep(self.rep[item])
            return self.rep[item]
    def check(self, u, v):
        return self.getrep(u) == self.getrep(v)
    def merge(self, u, v) :
        fu = self.getrep(u)
        fv = self.getrep(v)
        su = self.siz[fu]
        sv = self.siz[fv]
        if fu == fv :
            return True
        else :
            if su > sv :
                self.rep[fv], self.siz[fu] = fu, su + sv
            else :
                self.rep[fu], self.siz[fv] = fv, su + sv
            return False

N, K = map(int, input().split())
cnt = 0
D = DisjointSet(list(i for i in range(1, 3 * N + 1)))
for i in range(K) :
    op, u, v = map(int, input().split())
    if u > N or v > N:
        cnt += 1
        continue
    if op == 1:
        if D.check(u, v + N) or D.check(u, v + 2 * N) :
            cnt += 1
            continue
        D.merge(u, v)
        D.merge(u + N, v + N)
        D.merge(u + 2 * N, v + 2 * N)
    if op == 2:
        if D.check(u, v) or D.check(u, v + 2 * N) :
            cnt += 1
            continue
        D.merge(u, v + N)
```

```
            D.merge(u + N, v + 2 * N)
            D.merge(u + 2 * N, v)
    print(cnt)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



#45065855提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
class DisjointSet :
    def __init__(self, items) :
        self.rep = dict(zip(items, items))
        self.siz = dict(zip(items, [1] * len(items)))
    def getrep(self, item) :
        if self.rep[item] == item :
            return item
        else :
            self.rep[item] = self.getrep(self.rep[item])
            return self.rep[item]
    def check(self, u, v):
        return self.getrep(u) == self.getrep(v)
    def merge(self, u, v) :
        fu = self.getrep(u)
        fv = self.getrep(v)
        su = self.siz[fu]
        sv = self.siz[fv]
        if fu == fv :
            return True
        else :
```

基本信息

#: 45065855
题目: 01182
提交人: 23n2300010724
内存: 25824kB
时间: 851ms
语言: Python3
提交时间: 2024-05-24 16:00:45

2. 学习总结和收获

这次做作业的过程中发现Python好像很容易MLE的样子，这次积累的经验必将活用于下一次（笑）。