Assignment #F: All-Killed 满分

Updated 1844 GMT+8 May 20, 2024

2024 spring, Complied by 武昱达

编程环境

PyCharm 2023.1.4 (Professional Edition)

1. 题目

22485: 升空的焰火，从侧面看

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

思路:

打层数标记的BFS

代码

```
from collections import deque,defaultdict
N=int(input())
l=[-1 for _ in range(N+1)]
r=[-1 for _ in range(N+1)]
for i in range(1,N+1):
    left,right=map(int,input().split())
    l[i],r[i]=left,right

def BFS(root):
    q,step=deque(),0
    q.append((root,step))
    buffer,ans=defaultdict(list),[]

    while q:
        cur,step=q.popleft()
        buffer[step].append(cur)
        if l[cur]!=-1:q.append((l[cur],step+1))
        if r[cur]!=-1:q.append((r[cur],step+1))

    for step in buffer:
        ans.append(buffer[step][-1])
    return ans

ans=BFS(1)
print(*ans)
```

代码运行截图 （至少包含有"Accepted"）

#45027959提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque,defaultdict
N=int(input())
l=[-1 for _ in range(N+1)]
r=[-1 for _ in range(N+1)]
for i in range(1,N+1):
    left,right=map(int,input().split())
    l[i],r[i]=left,right

def BFS(root):
    q,step=deque(),0
    q.append((root,step))
    buffer,ans=defaultdict(list),[]

    while q:
        cur,step=q.popleft()
        buffer[step].append(cur)
        if l[cur]!=-1:q.append((l[cur],step+1))
        if r[cur]!=-1:q.append((r[cur],step+1))

    for step in buffer:
        ans.append(buffer[step][-1])
    return ans

ans=BFS(1)
print(*ans)
```

基本信息

#: 45027959
题目: 22485
提交人: 23n2300011119 (武)
内存: 3660kB
时间: 21ms
语言: Python3
提交时间: 2024-05-20 20:09:49

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

28203:【模板】单调栈

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

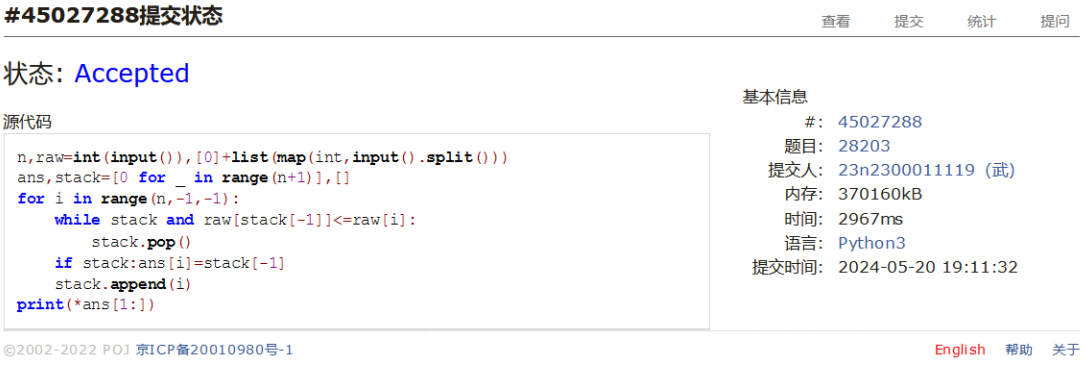
思路:

单调递减栈。

代码

```
n,raw=int(input()),[0]+list(map(int,input().split()))
ans,stack=[0 for _ in range(n+1)],[]
for i in range(n,-1,-1):
    while stack and raw[stack[-1]]<=raw[i]:
        stack.pop()
    if stack:ans[i]=stack[-1]
    stack.append(i)
print(*ans[1:])
```

代码运行截图 （至少包含有"Accepted"）



#45027288提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n,raw=int(input()),[0]+list(map(int,input().split()))
ans,stack=[0 for _ in range(n+1)],[]
for i in range(n,-1,-1):
    while stack and raw[stack[-1]]<=raw[i]:
        stack.pop()
    if stack:ans[i]=stack[-1]
    stack.append(i)
print(*ans[1:])
```

基本信息

#: 45027288
题目: 28203
提交人: 23n2300011119 (武)
内存: 370160kB
时间: 2967ms
语言: Python3
提交时间: 2024-05-20 19:11:32

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

09202: 舰队、海域出击!

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

思路:

Kahn算法拓扑排序

代码

```
from collections import deque
for _ in range(T:=int(input())):
    N,M=map(int,input().split())

    graph={i:[] for i in range(1,N+1)}
    in_degree=[0 for i in range(N+1)]

    for i in range(M):
        f,t=map(int,input().split())
        graph[f].append(t)
        in_degree[t]+=1

    q,cnt,visited=deque(),0,set()
    for i in range(1,N+1):
        if in_degree[i]==0:
            q.append(i)
    while q:
        cur=q.popleft()
        cnt+=1
        for vert in graph[cur]:
            in_degree[vert]-=1
            if in_degree[vert]==0 and vert not in visited:
                visited.add(vert)
                q.append(vert)

    print('No' if cnt==N else "Yes")
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

04135: 月度开销

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

思路:

经典二分法

代码

```
n,m=map(int,input().split())
expend=[int(input()) for i in range(n)]
def check(x):
    # 判断x作为最大月度开销是否可以实现，如果可以实现，则说明不够小或刚好符合题意。
    # 看m个方案是否可行。
    nums,s=1,0
    for i in range(n):
        if expend[i]+s>x:
            s=expend[i]
            nums+=1     # 求和大于设定的最大月度开销，则应该插入挡板，分份+1
        else:s+=expend[i]
    return nums>m   # if nums>m return True,else return False

lo,hi,res=max(expend),sum(expend)+1,1
while lo<hi:
    mid=(lo+hi)//2
    if check(mid):lo=mid+1
    else:res,hi=mid,mid

print(res)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#45028717提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n,m=map(int,input().split())
expend=[int(input()) for i in range(n)]
def check(x):
    # 判断x作为最大月度开销是否可以实现，如果可以实现，则说明不够小或刚好符合题意。
    # 看m个方案是否可行。
    nums,s=1,0
    for i in range(n):
        if expend[i]+s>x:
            s=expend[i]
            nums+=1      # 求和大于设定的最大月度开销，则应该插入挡板，分份+1
        else:s+=expend[i]
    return nums>m   # if nums>m return True,else return False

lo,hi,res=max(expend),sum(expend)+1,1
while lo<hi:
    mid=(lo+hi)//2
    if check(mid):lo=mid+1
    else:res,hi=mid,mid

print(res)
```

基本信息

#: 45028717
题目: 04135
提交人: 23n2300011119 (武)
内存: 7960kB
时间: 510ms
语言: Python3
提交时间: 2024-05-20 20:58:45

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

07735: 道路

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

思路:

```
# 重要剪枝：N个节点，不回头情况下至多N-1步。
if cost+nc<=K and step+1<N
```

代码

```
from heapq import heappop,heappush
from collections import defaultdict
K,N,R=int(input()),int(input()),int(input())
graph=defaultdict(list)
for i in range(R):
    S,D,L,T=map(int,input().split())
    graph[S].append((D,L,T))
def Dijkstra(graph):
    global K,N,R
    q,ans=[],[]
    heappush(q,(0,0,1,0))
    while q:
        l,cost,cur,step=heappop(q)
        if cur==N:return l
        for next,nl,nc in graph[cur]:
            if cost+nc<=K and step+1<N:
                heappush(q,(l+nl,cost+nc,next,step+1))
    return -1
print(Dijkstra(graph))
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#45032288提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from heapq import heappop,heappush
from collections import defaultdict
K,N,R=int(input()),int(input()),int(input())
graph=defaultdict(list)
for i in range(R):
    S,D,L,T=map(int,input().split())
    graph[S].append((D,L,T))
def Dijkstra(graph):
    global K,N,R
    q,ans=[],[]
    heappush(q,(0,0,1,0))
    while q:
        l,cost,cur,step=heappop(q)
        if cur==N:return l
        for next,nl,nc in graph[cur]:
            if cost+nc<=K and step+1<N:
                heappush(q,(l+nl,cost+nc,next,step+1))
    return -1
print(Dijkstra(graph))
```

基本信息

#: 45032288
题目: 07735
提交人: 23n2300011119 (武)
内存: 6988kB
时间: 42ms
语言: Python3
提交时间: 2024-05-21 12:44:30

01182: 食物链

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

思路:

一个并查集可以理解为带有两个虚部，非常巧妙。

代码

```python
class DisjointSet:
    def __init__(self, n):
        #设[1,n] 区间表示同类，[n+1,2*n]表示x吃的动物，[2*n+1,3*n]表示吃x的动物。
        self.parent = [i for i in range(3 * n + 1)] # 每个动物有三种可能的类型，用 3 * n 来表示每种类型的并查集
        self.rank = [0] * (3 * n + 1)

    def find(self, u):
        if self.parent[u] != u:
            self.parent[u] = self.find(self.parent[u])
        return self.parent[u]

    def union(self, u, v):
        pu, pv = self.find(u), self.find(v)
        if pu == pv:
            return False
        if self.rank[pu] > self.rank[pv]:
            self.parent[pv] = pu
        elif self.rank[pu] < self.rank[pv]:
            self.parent[pu] = pv
        else:
            self.parent[pv] = pu
            self.rank[pu] += 1
        return True


def is_valid(n,statements):
    dsu = DisjointSet(n)

```

```
    false_count = 0
    for d, x, y in statements:
        if x>n or y>n:
            false_count += 1
            continue

        if d == 1:  # 同类
            # 如果x和y中有任意一个未出现过，则二者之根（未出现过的那个的自身是其根）必然不相等，不存在冲突问题；
            # 如果x和y都出现过，则其必然已经分属三棵树，且y+n,y+2n,x+n,x+2n也已经分属三棵树。
            if dsu.find(x)==dsu.find(y+n) or dsu.find(x)==dsu.find(y+2*n):  # 不是同类，与条件矛盾
                false_count += 1
            else:
                dsu.union(x,y) # x,y是同种动物
                dsu.union(x+n,y+n) # x,y吃同一种动物
                dsu.union(x+2*n,y+2*n) # x,y被同一种动物吃
        else:  # X吃Y
            if dsu.find(x) == dsu.find(y) or dsu.find(x + 2*n) == dsu.find(y):  # 如果x,y是同类
                false_count += 1
            else: #[1,n] 区间表示同类，[n+1,2*n]表示x吃的动物，[2*n+1,3*n]表示吃x的动物
                dsu.union(x+n,y) # x吃的动物和y是同类
                dsu.union(x,y+2*n) # x和吃y的动物是同类
                dsu.union(x+2*n,y+n) # 吃x的动物和y吃的动物是同类

    return false_count


if __name__ == "__main__":
    N, K = map(int, input().split())
    statements = []
    for _ in range(K):
        D, X, Y = map(int, input().split())
        statements.append((D, X, Y))
    result = is_valid(N,statements)
    print(result)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态：Accepted

源代码

```
class DisjointSet:
    def __init__(self, n):
        #设[1,n] 区间表示同类，[n+1,2*n]表示x吃的动物，[2*n+1,3*n]表示吃x的动物
        self.parent = [i for i in range(3 * n + 1)] # 每个动物有三种可能的关
        self.rank = [0] * (3 * n + 1)

    def find(self, u):
        if self.parent[u] != u:
            self.parent[u] = self.find(self.parent[u])
        return self.parent[u]

    def union(self, u, v):
        pu, pv = self.find(u), self.find(v)
        if pu == pv:
            return False
        if self.rank[pu] > self.rank[pv]:
            self.parent[pv] = pu
        elif self.rank[pu] < self.rank[pv]:
            self.parent[pu] = pv
        else:
            self.parent[pv] = pu
            self.rank[pu] += 1
        return True


def is_valid(n,statements):
    dsu = DisjointSet(n)

    false_count = 0
```

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。

道路是一个很有趣的双指标BFS，但是题目输入数据有点坑；

食物链非常巧妙的并查集。

本次作业难度适中，但是需要刷题提高熟练度，迎接机考。