答案

1.（1）**8**　（2）**1**　（3）①**range(m)**　或　**range(m−1,−1,−1)**

或其他等价答案　②**k=task[k][1]**　或其他等价答案

③**w[k]=w[task[k][1]]−task[k][0]**　或

w[k]=w[c[k+1]]−task[k][0]　或其他等价答案

2.（1）**9**　（2）**data[i][2]=pr[num]**

（3）①**pr[loc]=data[pr[loc]][2]**

②**p!=−1 and data[p][1]>=data[vel][1]**　或

p!=−1 and data[p][1]>=power　③**data[vel][2]=p**

3.（1）**206**　（2）①**True**　②**199**　（3）①**idx+=1**

②**need<=num<bestnum**　③**lst p] 0]+=need**

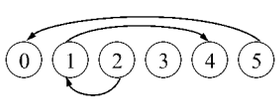
区块练 5 数据结构与算法的综合应用(一)

(核心考向:链表遍历、链表节点的插入与删除、链表排序)

1. 现真题 某工程包含n个任务(编号为0～n-1),每天可以有多个任务同时进行。某些任务之间有依赖关系,如图a所示,任务4依赖于任务1,任务1依赖于任务2,即任务2完成后才可以开始任务1,任务1完成后才可以开始任务4。不存在一个任务依赖于多个任务,或多个任务依赖于同一个任务的情况。

现已对该工程的依赖关系进行了梳理,结果如图b所示,标记"T"表示依赖关系需保留,标记"F"表示依赖关系需删除。

根据每个任务完成所需的天数和梳理后的依赖关系,编写程序,首先删除标记为"F"的依赖关系,然后计算工程最快完成所需的天数,并以工程最快完成所需的天数为期限,计算每个任务最晚必须开始的时间。



图a

任务A	任务B	标记
0	5	T
5	4	F
4	1	T
1	2	T
2	3	F

注:任务A依赖于任务B

图b

请回答下列问题:

(1)若某工程有6个任务,任务间依赖关系如图a所示,完成任务0～5所需天数分别为2,1,3,5,1,6,则工程最快完成需要_____天。

(2)定义如下erase(lst)函数,参数lst列表的每个元素表示一个依赖关系。函数的功能是删除标记为"F"的依赖关系,返回保留的依赖关系的个数。

```
def erase(lst):
    i=0;j=len(lst)-1
    while i<=j:
        if lst[i][2]=='T':
            i+=1
        else:
            if lst[j][2]=='T':
                lst[i]=lst[j];i+=1
            j-=1
    return i
```

若lst列表依次存储图b所示的依赖关系,如lst[0]为[0,5,'T'],调用erase(lst)函数,则语句"lst[i]=lst[j]"的执行次数为_____。

(3)实现上述功能的部分Python程序如下,请在划线处填入合适的代码。

```
def proc(n,lst,task):
    #w[i]存放任务i最晚必须开始的时间
    pr=[0]*n;w=[0]*n
    m=erase(lst)
    for i in _____①_____:
        task[lst[i][1]][1]=lst[i][0]
        pr[lst[i][0]]=1
    c=[]
    #days存放工程最快完成所需的天数
    days=0
    for i in range(n):
        if pr[i]==0:
            k=i;s=0
            while k!=-1:
                c.append(k)
                s+=task[k][0]
```

```
            ②
        if s>days:
            days=s
    for i in range(n-1,-1,-1):
        k=c[i]
        if task[k][1]==-1:
            w[k]=days-task[k][0]+1
        else:
            ③
    '''输出 days,以及保存在 w 中的每个任务最晚必须开始的时间,代码略'''
'''工程包含的任务数存入变量 n
任务间的依赖关系存入 lst 列表
lst[i]包含 3 项,任务 lst[i][0]依赖于任务 lst[i][1],lst[i][2]存放保留/删除标记,任务数据存入 task 列表
task[i]包含 2 项,task[i][0]为完成任务 i 所需天数,task[i][1]的初值为-1
代码略'''
proc(n,lst,task)
```

2.[2026 湖丽衢模拟]某未来社区设置 m 个共享电动车停靠点(编号 0～m-1),投放 n 辆共享电动车(编号 0～n-1)并平均分配至各停靠点,电动车电量范围为 0～100。用户登录系统后,可在停靠点借车,系统优先解锁该停靠点电量最高的电动车,电量相同时解锁其中闲置时间最长的。用户使用后归还至任意停靠点,系统记录归还车辆的剩余电量,当剩余电量低于 20 时,停靠点驻守的智能机器人立即更换电量为 100 的电池并投入使用。请回答下列问题:

(1)若某时刻 2 号停靠点和 3 号停靠点停放的车辆编号及其剩余电量如图 a 所示,系统已产生的借还记录如图 b 所示,则用户在 3 号停靠点借车的车辆编号为_____。

2 号停靠点		3 号停靠点	
车辆编号	剩余电量	车辆编号	剩余电量
1	44	3	27
4	97	8	73
2	34	6	100
7	65	9	80

图 a

记录顺序	操作类型	停靠点	车辆编号	剩余电量
1	借	2	4	97
2	借	3	6	100
3	还	3	4	80
4	还	3	6	77

图 b

(2)定义如下 classfiy(data,m)函数,列表 data 存储电动车两个数据项,编号为 i 的车辆的停靠点和剩余电量分别存储在 data[i][0]和 data[i][1]中。初始时所有车辆的电量均为 100,函数功能是将电动车按照其停靠点进行分类。

```
def classfiy(data,m):
    pr=[-1 for i in range(m)]
    for i in range(len(data)):
        data[i].append(-1)
        num=data[i][0]
        if pr[num]!=-1:
            _____
        pr[num]=i
    return pr
```

划线处应填入的语句是_____。

(3)系统实时记录用户的借车和还车数据,根据用户数据实时分配电动车并更新调整的部分 Python 程序如下,请在划线处填入合适的代码。

```
'''读取共享电动车数据,存入列表 data 中,读取停靠点数量存入 m 中,代码略'''
pr=classfiy(data,m)
while True:
    '''读取用户操作类型数据存入 t 中,0 表示借车,1 表示还车,代码略'''
```

```
    if t==0:
        '''读取用户的借车停靠点编号存入loc中,代码略'''
        if pr[loc]==-1:
            print("无车可借")
        else:
            print("借出车辆",pr[loc])
            _____①_____
    else:
        '''读取还车停靠点编号、车辆编号和剩余电量存入loc、vel和power中,代码略'''
        if power<20:
            #更换电池,代码略
            power=100
        data[vel][1]=power
        q=p=pr[loc]
        while _____②_____:
            q=p
            p=data[p][2]
        if q==p:
            pr[loc]=vel
        else:
            data[q][2]=vel
        _____③_____
```

3. 现真题 某中转站负责转运物品,每个物品规格相同且有唯一的正整数编号,物品送达中转站的时间先后与编号大小无关,每次运走的物品编号均需连续。

将中转站中的现有物品编号划分为若干序列,连续的编号按升序置于同一序列,不连续的编号置于不同序列。当需要运走x个物品时,检查当前时间现有序列中是否存在长度≥x的序列,若存在,则:

①选择长度最短的序列;若最短序列有多个,则选择起始编号最小的序列。

②选定序列后,运走从其起始编号开始的x个连续编号的物品。

若不存在符合条件的序列,则本次需求取消。请回答下列问题:

(1)若物品按如图所示的时间送达中转站,第一次需要运走的物品个数为3,时间为09:15,则该次运走物品的起始编号为_____。

时间	08:15	08:20	08:30	08:55	09:00	09:05	09:10	09:25
编号	202	206	203	201	208	207	204	205

(2)定义如下函数check(reqs,m),用于判断列表reqs中是否存在下列情况:任意连续60分钟内,需要运走的物品数量总和超过m。列表reqs中的每个元素包含两个数据项,依次为需要运走的时间(已转换为分钟数)和物品数量,已按时间升序排列。

```
def check(reqs,m):
    i=j=0
    s=0
    while j<len(reqs):
        if reqs[j][0]-reqs[i][0]<60:
            s+=reqs[j][1]
            if s>m:
                return True
            j+=1
        else:
            s-=reqs[i][1]
            i+=1
    return False
```

调用该函数,请回答①和②两个问题。

①若reqs为[[6,1],[19,13],[35,3],[70,2],[75,7]],m为20,则返回值为_____。

②若reqs的长度为100,函数返回值为False,则while语句中循环体的执行次数最多是_____次。

(3)中转站需逐条处理需求信息,根据物品的送达时间和编号,以及需要运走的时间和

数量，记录成功运走的物品编号，模拟上述过程的Python程序如下，请在画线处填入合适的代码。

```
def addpid(pid,lst):
    '''
    该函数根据新加入的物品编号pid更新链表。
    列表lst模拟链表结构，链表节点的前两个数据项依次为连续编号序列的起始和终止编号，第三个数据项为指针。lst[0]为链表的头节点（前两个数据项不存储有效数据），其余节点在链表中按起始编号升序排列，连续编号序列存于同一节点。
    例如：若pid为19，lst为[[-1,-1,2],[15,18,3],[11,13,1],[20,22,-1]]，调用addpid(pid,lst)后，lst为[[-1,-1,2],[15,22,-1],[11,13,1],[20,22,-1]]
    '''
    #代码略
def proc(reqs,items):
    res=[]
    #初始化为仅包含头节点的链表
    lst=[[-1,-1,-1]]
    idx=0
    for i in range(len(reqs)):
        cur,need=reqs[i][0],reqs[i][1]
        while idx<len(items) and items[idx][0]<cur:
            pid=items[idx][1]
            '''根据新加入的物品编号pid更新链表'''
            addpid(pid,lst)
            ①
        bestnum=len(items)+1
        bestpre,pre=-1,0
        p=lst[0][2]
        while p!=-1 and need!=bestnum:
            num=lst[p][1]-lst[p][0]+1
            if ②:
                bestnum=num
                bestpre=pre
            pre=p
            p=lst[p][2]
        if bestpre!=-1:
            p=lst[bestpre][2]
            #为res追加一个元素
            res.append([i,lst[p][0],lst[p][0]+need-1])
            if bestnum==need:
                lst[bestpre][2]=lst[p][2]
            else:
                ③
        else:
            #取消本次需求，代码略
    return res
'''
将物品信息存入items列表，每个元素包含两个数据项，依次为物品送达时间和编号；
将需求信息存入reqs列表，每个元素包含两个数据项，依次为需要运走的时间和数量；
两个列表均已按时间升序排列。
代码略
'''
res=proc(reqs,items)
#输出res列表中的处理结果，代码略
```