高三下信息选考算法综合练习 6

[26.4 嘉兴二模]1.某快递服务站按以下规则排序包裹：加急件优先配送，同类型包裹重量大的优先。每个包裹信息格式为 [快递单号, 重量(kg), 类型(“急”/“普”)]。实现排序的 Python 程序如下：

```
a=[["A01", 2.0, "普"], ["B12", 1.5, "急"], ["C35", 3.2, "普"], ["D77", 2.8, "急"], …] n=len(a)
for i in range(n-1):
    for j in range(____①____):
        if a[j][1] < a[j+1][1] and a[j][2]==a[j+1][2]:
            a[j],a[j+1]=a[j+1],a[j]
        elif ____②____:
            a[j], a[j+1]=a[j+1],a[j]
```

为完成元素的排序，①②处代码依次为　D

A. ①n-1, i, -1 ②a[j][2]=="急" and a[j-1][2]=="普"
B. ①n-1, i, -1 ②a[j][2]=="普" and a[j-1][2]=="急"
C. ①n-i-1 ②a[j][2]=="急" and a[j+1][2]=="普"
D. ①n-i-1 ②a[j][2]=="普" and a[j+1][2]=="急"

[26.4 嘉兴二模]2.某连锁超市每小时统计营业额，需找出连续 3 小时的最高营业额。已知 7 小时营业额为 money = [980, 720, 900, 850, 770, 950, 800]，连续 3 小时最高营业额依次为：980、900、900、950、 950。实现该功能的程序段如下：

```
money = [980, 720, 900, 850, 770, 950, 800]
n = len(money); index= [0] * 100
p1 = 0; p2 = 0; k = 3
ans = []
for i in range(n):
    if p1 != p2 and index[p1] <= i - k:
        ____①____
    while p1 != p2 and money[index[p2 - 1]] <= money[i]:
        ____②____
    ____③____
    p2 += 1
    if i >= k - 1:
        ans.append(money[index[p1]])
print("每连续 3 小时的最高营业额为：", ans)
```

划线处依次填入的代码为　B

A. ①p1 += 1　②p2 -= 1　③index[p2] = index[i]
B. ①p1 += 1　②p2 -= 1　③index[p2] = i
C. ①p2 += 1　②p1 -= 1　③index[p2] = index[i]
D. ①p2 += 1　②p1 -= 1　③index[p2] = i

总结：滑动窗口最大值（单调队列）

1. 队首元素：始终是当前窗口的最大值的索引。
2.队列内元素：对应的值呈非严格递减，队尾元素一定小于队首元素。
3.窗口移动时的三步操作：
①队首元素超出窗口范围时，从队首移除。
②新元素入队前，把队尾所有 ≤ 它的元素移除（它们不可能再成为后续窗口的最大值）。
③窗口形成后，每次取队首元素作为当前窗口的最大值

[26 年 4 月嘉兴二模]3.现有 n 种编号为 $1 \sim n$ 的商品，每种数量若干，需放入货架。每个货架最多放 m 个商品，货架编号从 1 开始且数量充足。入库规则如下：

第一轮：按商品编号从小到大的顺序处理每一种商品。如果某种商品的数量≥ m 个，就把 m 个商品放满一个货架；如果还能再放满一个货架，就继续放。当剩余数量不足 m 个时暂不放置，留到第二轮处理。

第二轮：按“剩余商品数量从大到小”排序（数量相同则编号小的优先），依次放置剩余商品：①优先放入已有货架中：剩余空位≥该商品剩余数量，且空位最少、编号最小的货架；②若无满足条件的已有货架，则放入新货架。

例：现有 3 种编号的商品需入库，每个货架最多能放 15 个商品，各编号商品数量及放置过程如下：

商品编号	商品数量
1	19
2	10
3	20

商品编号	剩余数量	货架编号
1	4	1
2	10	
3	5	2

商品编号	剩余数量	货架编号
2	10	3
3	5	2，3
1	4	1，4

（1）若仅将示例中编号 1 的商品数量改为 39，则编号为 3 的商品放置的货架编号依次为 3,4 。（货架编号间用逗号分隔）

（2）定义如下 lninsert(x, head) 函数，功能为将商品信息 x 插入到链表 lst 中，使得链表按商品剩余数量降序排列。lst 节点的前两个数据项依次为商品编号和剩余数量，第三个数据项为指针。x 为列表，其中 x[0]为商品编号，x[1]为商品剩余数量，head 为链表头的指针。为实现上述功能，请在程序划线处填入合适的代码。

```
def lninsert(x, head):
    if head == -1:
        lst.append([x[0], x[1], -1])
        head = 0
    elif x[1] > lst[head][1]:
        lst.append([x[0], x[1], head])
        head = len(lst) - 1
    else:
        p = q = head
        while q != -1 and x[1] < lst[q][1]:
            p = q
            q = lst[q][2]
        lst.append([x[0], x[1], q])
        lst[p][2]=len(lst)-1
    return head
```

（3）实现商品放置功能的部分 Python 程序如下，请在程序中划线处填入合适的代码。

```
def place(goods, m):
    n = len(goods)
    result = []  # 存放各类商品所放置的货架编号
    for i in range(n + 1):
        result.append([])
    rest = [m] * 101  # 假设货架数不超过 100 个
    hjbh = 0
    head = -1
    for i in range(n):
        while goods[i][1] >= m:
            hjbh = hjbh + 1
            result[goods[i][0]].append(hjbh)
            goods[i][1]-=m
        if goods[i][1] > 0:
            head = lninsert(goods[i], head)
    start = hjbh + 1
```

```
    p = head
    while p != -1:
        cnt = lst[p][1]
        empty = m
        for j in range(start, hjbh + 1):
            if ____②____:
                empty = rest[j]
                ans = j
        if empty == m:
            ____③____
            ans = hjbh
        result[lst[p][0]].append(ans)
        rest[ans] -= lst[p][1]
        p = lst[p][2]
    return result
# 主程序
# 列表 goods 存储编号 1 至 n 的商品数，依次存入 goods[0]至 goods[n-1]中;
#goods[i]包含 2 个数据项，goods[i][0]，goods[i][1]分别存放商品编号及数量。
lst = []
m = int(input("m="))
goods = [[1, 31], [2, 17], [3, 11], [4, 40]]
print(place(goods, m))
```

rest[j]>=cnt and rest[j]<empty

hjbh+=1

[26.4 金华十校]4.外卖推荐系统会根据每家外卖店的订单信息，动态调整系统中的优先推荐名单，具体规则如下：

①外卖店用 1~N 编号，初始时刻每家外卖店的优先级都为 0。

②每经过 1 个时间单位，如果外卖店没有订单，则优先级会减少 1，最低减到 0；而如果外卖店有订单，则每有一单优先级会加 2。

③如果某家外卖店某时刻优先级大于 5，则会被系统加入优先推荐名单；如果优先级小于等于 3，则会被清除出优先推荐名单。

例如有 3 家外卖店，在 6 个单位时间里面的订单信息如第 15-1 题图所示，则到时刻 6 时店铺 2 在优先推荐名单中，外卖推荐系统的动态调整过程如第 15-2 题图所示。

时刻 t	店铺 id
1	1
5	2
3	1
6	2
4	3
3	2
6	2
2	1

第 15-1 题图

店铺 id	时刻 0	时刻 1	时刻 2	时刻 3	时刻 4	时刻 5	时刻 6
1	0	2	4	6(推荐)	5(推荐)	4(推荐)	3
2	0	0	0	2	1	3	7(推荐)
3	0	0	0	0	2	1	1

第 15-2 题图

时刻 t	店铺 id
2	3
1	3
3	3
6	2
4	2
3	1
6	3
2	1
5	2

第 15-3 题图

(1)若 3 家店铺的订单信息如第 15-3 题图所示，则时刻 6 时优先推荐名单中有____▲2____家店铺。

(2)对订单排序的 bsort 函数如下：

```
def bsort(data): # 列表 data 保存订单数据,每个订单数据为[时刻 t,店铺 id]
    n = len(data)
    c = 0
    for i in range(n - 1):
        for j in range(n - 1 - i):
            if data[j][0] > data[j+1][0]:
                data[j], data[j+1] = data[j+1], data[j]
            elif data[j][0] == data[j+1][0] and data[j][1] > data[j+1][1]:
                data[j], data[j+1] = data[j+1], data[j]
                c += 1
```

①bsort 函数运行后,data 中的订单数据排序规则是__▲ C__(单选)
A. 按时刻 t 从小到大排序;时刻相同时,按店铺 id 从大到小排序
B. 按店铺 id 从小到大排序;id 相同时,按时刻 t 从小到大排序
C. 按时刻 t 从小到大排序;时刻相同时,按店铺 id 从小到大排序
②若 data 中的数据如第 15-3 题图所示,则运行完 bsort 函数后,c 的值是__▲ 2__。
(3)外卖推荐系统按订单数据逐个处理,计算并以店铺 id 升序输出优先推荐店铺名单,请在划线处填入合适的代码。

```
#n, m, T 分别保存店铺数量,订单数量,结束时刻,代码略
# 列表 data 保存具体的订单数据,每个订单数据存储为[时刻 t,店铺 id],代码略
bsort(data)
s = []
st = [False] * (n + 1)
val = [0] * (n + 1)
post = [0] * (n + 1)
i = j = 0
while i < m:
    while j < m and data[i] == data[j]:
        j += 1
    cnt = j - i
    tt , sid = data[i][0] , data[i][1]
    val[sid] -= ①tt-post[sid]-1
    post[sid] = tt
    if val[sid] < 0:
        val[sid] = 0
    val[sid] += cnt * 2
    if val[sid] > 5:
        if ②st[sid]==False:
            st[sid] = True
            s.append(sid)    # 将 sid 添加到列表 s 的末尾
            k = len(s) - 1
            while s[k] < s[k - 1] and k >= 1:
                s[k - 1], s[k] = s[k], s[k - 1]
                k -= 1
    i = j
print("优先推荐店铺名单:")
for sid in s:
    if post[sid] < T:
        val[sid] -= T - post[sid]
    if ③val[sid]>3:
        print("店铺", sid)
```