

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

- 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

- 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 \leq 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 $\geq m$ 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 \geq 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_____。（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])
        _____
    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)
            _____
        if goods[i][1] > 0:
            head = lninsert(goods[i], head)
    start = hjbh + 1
```


①bsort 函数运行后, data 中的订单数据排序规则是 ▲_____ (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▲_____。

(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] -= _____ ①
```

```
    post[sid] = tt
```

```
    if val[sid] < 0:
```

```
        val[sid] = 0
```

```
    val[sid] += cnt * 2
```

```
    if val[sid] > 5:
```

```
        if _____ ②:
```

```
            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 _____ ③:
```

```
        print("店铺", sid)
```