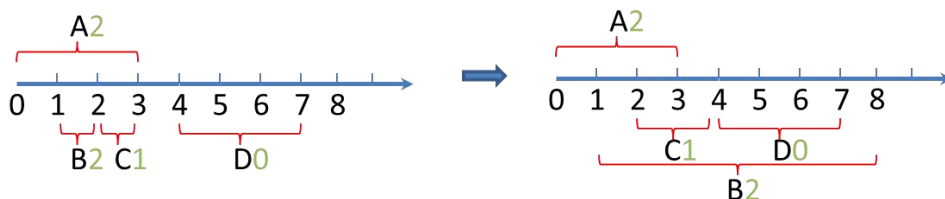
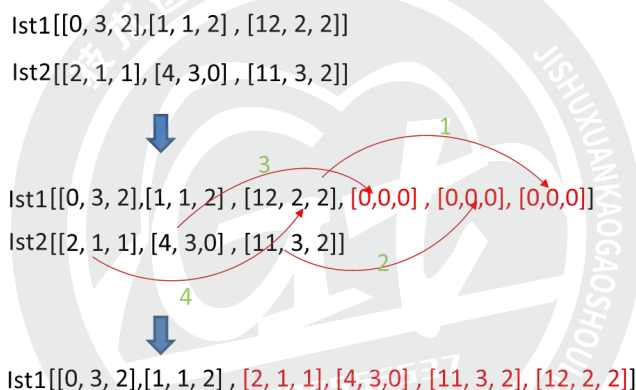


本题考查基于问题解决的程序综合阅读能力，涉及的知识点有程序基础、自定义函数、双指针、链式队列和顺序查找算法等。

(1) 从表格中可以看出 A 的优先级为 2，A 在处理时，B 和 C 在等待，A 在时刻 3 处结束，B 和 C 等待了 2 秒和 1 秒，C 的优先级比 B 的优先级高，C 先处理，从时刻 2 到达，在时刻 4 结束。D 在时刻 4 到达，优先级比 B 高，D 先处理，在时刻 7 结束，因此 B 在 7 时刻开始处理，每个器件等待时长为其开始检测的时间与送达时间的的时间差，因此 B 的等待时长为 $7-1=6$ 。如图所示：



(2) ①若 $lst1$ 为 $[[0, 3, 2], [1, 1, 2], [12, 2, 2]]$ ， $lst2$ 为 $[[2, 1, 1], [4, 3, 0], [11, 3, 2]]$ ，结合自定义函数的程序，将具体数据代入，程序先在 $lst1$ 后添加 $\text{len}(lst2)$ 个数据元素，再比较大小，将较大值从后往前放入 $lst1$ 中，过程如下图所示，可知循环体执行了 4 次。



②若函数中 `while` 语句的条件“ $j \geq 0$ ”误写为“ $k \geq 0$ ”， j 是指向 $lst2$ 的指针，则当 $lst2$ 中的数据已经处理完时，会出问题，因此答案选 A。

(3) 解决问题的方式采用链表实现的队列，即链式队列，需要 2 个指针来控制（头和尾），二维列表 `queinfo` 就用来存储每个优先级的头尾指针。`data` 追加一个元素 -1 用来存储下一个处理的指针。

整段自定义函数要计算平均等待时间，从 total/n 可以看出，程序需要通过 $\text{total} += \text{curtime} - \text{data}[p][0]$ 累计 total 值，因此①处答案为 $\text{total} = 0$ 。

核心程序循环体部分框架比较明显，第 1 个条件 $i < n$ and $\text{data}[i][0] \leq \text{curtime}$ 成立的情况下，执行入队操作，第 2 个条件 $\text{waitnum} > 0$ 成立的情况下执行出队操作，最后 `else` 部分执行更新 curtime 操作。

以数据 $\text{data} = [[1, 5, 1], [10, 25, 3], [20, 13, 0], [21, 9, 0], [22, 2, 0], [23, 5, 2], [24, 2, 0]]$ 为例子，第 1 个器件是时刻 1 到达的，先进入第 3 个条件，更新 curtime 为 1，然后进入第 1 个条件，根据优先级 1，将器件索引 0 入队，即 $\text{queinfo}[1]$ 更新为 $[0, 0]$ ，接着进入第 2 个条件进行处理，该器件的等待时间为 0， curtime 更新为该器件处理完的时刻 6，处理即出队操作，因为队列中没有其他等待器件索引，因此 $\text{queinfo}[1][0]$ 更新为 -1，最终 queinfo 更新为 $[-1, -1], [-1, 0], [-1, -1], [-1, -1]]$ 。

处理第 2 个器件时， curtime 更新为 10，先入队，即 $\text{queinfo}[3]$ 更新为 $[1, 1]$ ，再出队， $\text{queinfo}[3][0]$ 更新为 -1， queinfo 更新为 $[-1, -1], [-1, 0], [-1, -1], [-1, 1]]$ ， curtime 更新为时刻 35，该器件的等待时间为 0。

后面器件时入队时前面器件还没处理完，入队等待，第 3 个器件到达时 $\text{queinfo}[0]$ 更新为 $[2, 2]$ ，第 4 个器件到达时 $\text{queinfo}[0]$ 更新为 $[2, 3]$ ，第 5 个器件到达时 $\text{queinfo}[0]$ 更新为 $[2, 4]$ ，第 7 个器件到达时 $\text{queinfo}[0]$ 更新为 $[2, 6]$ ，队列 $\text{queinfo}[0]$ 中存储的是优先级为 0 的器件的头尾索引，此过程通过 $\text{queinfo}[k][1] = i$ 实现，在此之前需要更新 data 中最后一个元素（存储指针区域），让其指向下一个节点，由 $\text{data}[p][3] = i$ 推理②

处答案为 $p = \text{queinfo}[k][1]$ 。

出队时， $[2, 6]$ 依次更新为 $[3, 6]$ 、 $[4, 6]$ 、 $[6, 6]$ 、 $[-1, 6]$ ，因此③处答案为 $\text{queinfo}[k][0] = \text{data}[p][3]$ 。

【解析二】（高手联盟 IT 组 提供）

(1) 本题比较简单，考查对题意的理解。

结合题意“等待时长”是指开始检测的时间与送达时间的的时间差，本题中有数据 A、C、D 作参考，可以帮助学生加深题意的理解，对后续解题过程起到一个衔接作用。器件 A、C、D、B 具体检测时间安排，如下图所示。



器件	开始检测时间	结束检测时间	等待时长
A	0	3	(送检时间 0) 0
C	3	4	(送检时间 2) $3-2=1$
D	4	7	(送检时间 4) 0
B	7	8	(送检时间 1) $7-1=6$

(2) ①题相对简单，考查学生对二路归并算法思想的了解。当 lst2 中的数据全部完成归并后，即 while 循环结束。因此我们将观察点放置在 lst2 中的数据是否已经完成归并，结合二路归并算法思想，归并过程中的每次数据选择如下：

lst1 为 $[[0, 3, 2], [1, 1, 2], [12, 2, 2]]$

lst2 为 $[[2, 1, 1], [4, 3, 0], [11, 3, 2]]$

0	1	2	3	4	5	说 明
					$[12, 2, 2]$	$i=2, j=2$, 选择 $\text{lst1}[5]=\text{lst1}[2](i=1)$
				$[11, 3, 2]$	$[12, 2, 2]$	$i=1, j=2$, 选择 $\text{lst1}[4]=\text{lst2}[2](j=1)$
			$[4, 3, 0]$	$[11, 3, 2]$	$[12, 2, 2]$	$i=1, j=1$, 选择 $\text{lst1}[4]=\text{lst2}[1](j=0)$
		$[2, 1, 1]$	$[4, 3, 0]$	$[11, 3, 2]$	$[12, 2, 2]$	$i=1, j=0$, 选择 $\text{lst1}[4]=\text{lst2}[0](j=-1)$

while 语句中的循环体执行 4 次后，变量 j 值为 -1， lst2 中数据全部完成归并，循环结束。

本题二路归并思想，解析如下：

```
def merge(lst1, lst2):
    i = len(lst1) - 1          # 升序序列 lst1, 数据区间[0..i]
    j = len(lst2) - 1          # 升序序列 lst2, 数据区间[0..j]
    for t in range(len(lst2)):  # 二路归并后，数据区间是：[0..len(lst1+lst2)-1]
        lst1.append([0, 0, 0])
    k = len(lst1) - 1
    # 每次从原 lst1, lst2 中选择最大元素，将其放在 k 位置，对 lst1 中未处理数据不造成影响
    while j >= 0:               # 当 lst2 中还有未归并的数据时进行处理
        # 当 lst1 中有数据，且 lst1 中数据大于 lst2 数据时，选择 lst1 中数据归并
        if i >= 0 and lst1[i][0] > lst2[j][0]:
            lst1[k] = lst1[i]
            i -= 1
        else:                    # 选择 lst2 中数据做归并
            lst1[k] = lst2[j]
            j -= 1
        k -= 1
    return lst1
```

②本题有一定的难度，从测试数据角度考查学生代码调试能力，考察学生对二路归并算法的理解。 while

循环变量的不同，可从不同角度体现归并算法差异。具体如下：

```
# 每一次从 lst1, lst2 中选择最大元素，将其放在 k 位置
k=len(lst1)-1
while k>=0:          # 变量 k 可看成做归并时，总数据的个数
    if i<0 or i>=0 and lst1[i][0]>lst2[j][0]: # ① 选择 lst1 中的数据归并
        lst1[k]=lst1[i]
        i-=1
    else:             # ② 选择 lst2 中数据归并
        lst1[k]=lst2[j]
        j-=1
    k-=1
```

若 while 循环条件修改为 k 后，当 lst2 中数据完成归并时变量 j 值为 -1，lst1[i] 的比较对象是 lst2[-1] (lst2 中最后的数据)，选项 A 数据存在隐蔽错误，运行结果不正确。

选项 B、C、D 中 3 组测试数据均是 lst1 先完成归并，不会出现该错误，程序运行结果正确，因此选项 A 这组数据可以检测出该错误，答案选 A。

(3) ① total = 0 本题比较简单，找到计算总时间的变量 total，设置初始值为 0。

(3) ② p=queinfo[k][1] # 找到 k 等级链表的表尾

若该等级已存在其他器件，由于器件是按时间升序遍历。因此我们将该器件添加到 k 等级链表表尾。通过访问 k 等级虚点对应的链表表尾，找到表尾位置 p(p=queinfo[k][1])，然后在链表表尾追加当前器件的索引位置 i，完成待处理器件的入队操作。

(3) ③ queinfo[k][0]=data[p][3]

在 k 等级链表中，找到最高 k 等级单链表指向的位置 p，p 为单链表中队首器件位置。

此处是将 p 指向的器件删除，通过更新 k 等级链表虚点表头 queinfo[k][0]，使链表虚点表头指向 p 的下一个器件位置，完成删除操作，答案为 queinfo[k][0]=data[p][3]。

【代码解析】

```
def proc(data,m):
    n=len(data)
    queinfo=[]
    # 为每个不同等级创建虚点单链表(虚点中存储节点位置)
    # queinfo 追加[-1,-1](k 等级单链表表头位置:queinfo[k][0],表尾位置 queinfo[k][1])
    for i in range(m):
        queinfo.append([-1,-1])
    # data 中存放数据按时间升序排列,追加数据目的是为了更方便创建单链表 Next[i]指针
    for i in range(n):
        data[i].append(-1)
    curtime=0
    total=0          # 存储总等待时间
    waitnum=0        # 存在尚未处理的数据
    i=0              # i 指向当前待处理的器件
    while i<n or waitnum>0:
        # 若当前器件开始检测时间不早于 curtime，则将其放到对应等级的链表队列中
        if i<n and data[i][0]<=curtime:
            k=data[i][2]          # 当前器件等级为 k
            if queinfo[k][0]==-1: # 若 k 等级链表不存在
```

```

        queinfo[k][0]=i                # 将器件索引位置 i 设置为单链表的表头
    else:
        # 将器件索引位置 i 添加在 k 等级单链表的表尾
        # 相同 k 等级器件按开始检测时间入队
        p=queinfo[k][1]
        data[p][3]=i
        queinfo[k][1]=i                # 完成上面操作后,更新 k 等级单链表表尾
        waitnum+=1                     # 待处理器件个数增 1
        i+=1                           # 为下一个器件做准备
    elif waitnum>0:                     # 处理当前 k 等级队列中的元素
        k=0                            # k 等级单链表优先次序是: 0..m-1
        # 按 k 等级单链表的等级从高到低进行遍历, 找到一个最高等级非空单链表
        while queinfo[k][0]==-1:
            k+=1
        # 通过 k 等级虚点 queinfo[k][0], 找到 k 等级单链表表头元素索引位置 p
        p=queinfo[k][0]                # p 指向 k 等级队列的队首器件
        total+=curtime-data[p][0]      # 统计总等待时长
        curtime+=data[p][1]            # 更新当前时间 curtime
        # 将指针 p 指向的节点删除,让 k 等级单链表虚点指向 p 的下一个节点位置
        # k 等级队首器件出队
        queinfo[k][0]=data[p][3]
        waitnum-=1
    else:
        curtime=data[i][0]             # 不用等待,仅更新 curtime
    return total/n                     # 返回平均等待时间

```

技术选考高手联盟
JISHUXUANKAOGAOSHOU LIANMENG