

### 高三下信息选考练习卷 7

一、选择题(本大题共 12 小题,每小题 2 分,共 24 分。每小题列出的四个备选项中只有一个是符合题目要求的,不选、多选、错选均不得分)

阅读下列材料,回答第 1 至 6 题:

某交警支队的智慧信息系统可获取由摄像头抓拍的车辆违章图片及视频,自动识别车牌、违章行为并生成图文并茂的报告。经交警审核确认后,系统服务器将违章处理通知以短信的形式发送至车主手机。若车主对此存在异议,可通过客户端 App 进行反馈,系统自动推送至人工复核。管理员可查看系统识别准确率报表。

1. 下列关于该系统中数据的说法,正确的是 **B**

- A. 该系统中的所有数据都是结构化数据
- B. 存储在服务器中的数据均已完成数字化处理
- C. 经审核确认后的数据不具备价值
- D. 在数据处理过程中不会有新的数据产生

2. 摄像头以分辨率 1920 × 1080 像素,颜色位深度 24 位的参数抓拍,抓拍后压缩存储的某图像容量约为 1.5MB,则该图像压缩比为 **A**

- A. 4:1
- B. 8:1
- C. 16:1
- D. 32:1

3. 下列关于该系统组成的描述,不正确的是 **C**

- A. 系统的硬件包含摄像头
- B. 系统中的客户端 App 属于应用软件
- C. 车辆违章图片不属于系统的数据
- D. 车主和管理员都是系统的用户

4. 下列关于该系统安全与防护的做法,不合理的是 **D**

- A. 为系统配置防火墙
- B. 定期对系统数据进行备份
- C. 升级服务器杀毒软件
- D. 为不同的用户设置相同的操作权限

5. 下列关于该系统网络技术的分析,正确的是 **D**

- A. 系统的软件开发框架仅使用 B/S 架构
- B. 系统仅可通过局域网实现网络通信
- C. 车主的反馈数据上传至服务器无需遵循网络协议
- D. 管理员查看报表是网络资源共享的体现

6. 该系统识别行为基于神经网络方法实现,为提升系统识别准确率,下列措施不合理的是 **D**

- A. 完善系统对各类交通违章样本数据的训练
- B. 优化识别的算法
- C. 更换像素更高的摄像头
- D. 增加服务器的内存容量

7. 某算法的部分流程图如第 7 题图所示,若 n 的值为 6,数组元素 a[0]至 a[n-1]依次存放 2, 8, 4, 8, 6, 2, 执行这部分流程后,输出 k 的值为 **C**

- A. 0
- B. 1
- C. 3
- D. 5

8. 队列 queA 从队首到队尾的元素依次为 1, 2, 3, 4, 队列 queB 从队首到队尾的元素依次为 5, 6, 7。约定: T 操作是指元素从 queA 出队, U 操作是指元素从 queA 出队后至 queA 再入队, H 操作是指元素从 queB 出队后至 queA 再入队。经过 TUHTUH 系列操作后,队列 queA 的队首元素为 **A**

- A. 2
- B. 4
- C. 5
- D. 6

9. 某二叉树的树形结构如第 9 题图所示,其后序遍历结果为 DFEBCA, 则中序遍历结果为 **C**

- A. ABCDEF
- B. ABDEFC
- C. DBEFAC
- D. DBFEAC

10. 有如下 Python 程序段:

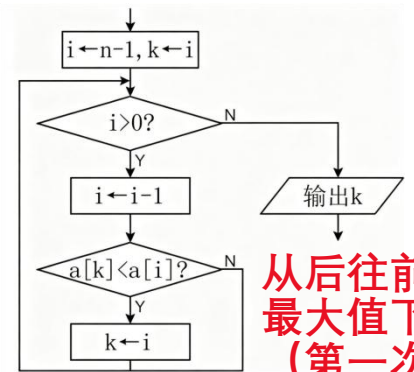
```
n=0
for c in s:
    if n==0:
        x=c
    if x==c:
        n+=1
    else:
        n-=1
```

若 s 为“甲丙乙丙丙丙丁”, 执行该程序段后, x 的值为 **C**

- A. “甲”
- B. “乙”
- C. “丙”
- D. “丁”

11. 有如下 Python 程序段:

```
n=len1+len2
i=0;j=len1
```

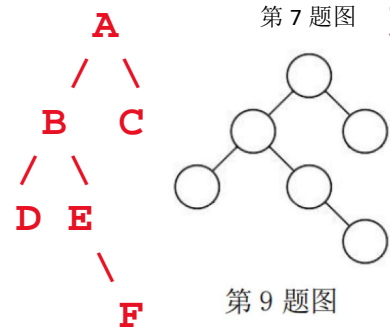


第 7 题图

从后往前找  
最大值下标  
(第一次出现)

字母消消乐:

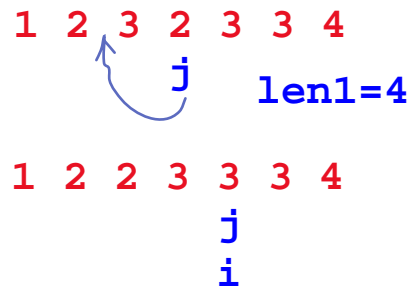
~~甲~~ ~~丙~~ ~~乙~~ ~~丙~~ ~~丙~~ ~~丁~~



第 9 题图

## 将a[j]插入到前面

```
while i < len1 and j < n:
    if a[i] > a[j]:
        tmp = a[j]
        for k in range(j, i, -1):
            a[k] = a[k-1]
        a[i] = tmp
        len1 += 1
        j += 1
    i += 1
```



若 a 为 [1, 2, 3, 2, 3, 3, 4], len1 为 3, len2 为 4, 执行该程序段后, i 的值为 **B**

A. 3                      B. 4                      C. 5                      D. 6

```
12. 有如下 Python 程序段:
from random import randint
n = len(a); stk = [0]*n; top = -1
s = 0
```

```
for i in range(n):
    if a[i] == -1:
        if randint(0, 1) == 1 and top != -1: # randint(0, 1) 随机生成 0 或 1
            s += stk[top]                      出栈
            top -= 1
        else:
            top += 1
            stk[top] = a[i]
```

**s = 1 + 2 + 16 + 8**  
**s = 1 + 16 + 8**  
**s = 1 + 16**  
**s = 2 + 16 + 8**  
**s = 2 + 16**  
**s = 16 + 8**  
**s = 16**  
 .....

**1, -1, 2, -1, 4, 8, 16, -1, -1**  
**s    +1       +2                      +16   +8**

若 a 为 [1, -1, 2, -1, 4, 8, 16, -1, -1], 执行该程序段后, s 的值可能为 **A**

A. 25 = **1 + 16 + 8**    B. 21                      C. 10                      D. 7

## 二、综合题(本大题共 3 小题, 其中第 13 小题 10 分, 第 14 小题 7 分, 第 15 小题 9 分, 共 26 分)

13. 某小组模拟搭建苗圃大棚环境监测系统, 采用智能终端连接湿度传感器、光线传感器, 每分钟采集一次湿度和光照数据, 并通过 Wi-Fi 将数据传输至服务器, 存储到数据库中。服务器处理数据后, 通过智能终端控制加湿器等设备运行。用户可通过浏览器查询实时和历史数据。请回答下列问题:

(1) 该系统中数据流向不合理的是 **▲ A** (单选)。

- A. 智能终端 ← 执行器                      B. 智能终端 ↔ 服务器                      C. 服务器 ↔ 浏览器

(2) 编写智能终端程序时, 需要知道 **▲ ABC** (多选) (注: 全部选对的得 2 分, 选对但不全的得 1 分, 不选或有错选的得 0 分)

- A. 与传感器连接的智能终端引脚                      B. Wi-Fi 的 SSID 及密码  
 C. 服务器的地址及端口                      D. 数据库的文件名

(3) 下列关于该系统的说法, 正确的有 **▲ BD** (多选)。(注: 全部选对的得 2 分, 选对但不全的得 1 分, 不选或有错选的得 0 分)

- A. 传感器和执行器不能连接至同一智能终端  
 B. 可以基于 Flask Web 框架编写服务器端程序  
 C. 服务器负责所有的数据存储, 智能终端负责所有的数据处理  
 D. 通过浏览器查看系统历史数据需要访问数据库

## 目的: 减少数据

(4) 系统运行后, 数据库逐渐庞大, 在不影响系统功能且不更改任何硬件的前提下, 为减轻服务器的存储压力, 请写出 1 种合理的做法。 **定期将数据进行异地转存, 适当增大数据的采集时间间隔**

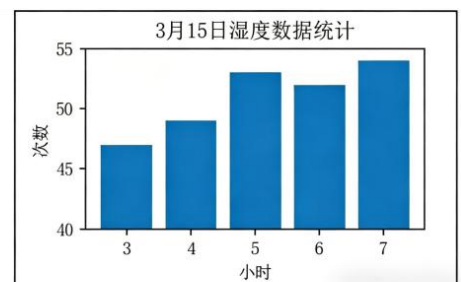
(5) 将当年 3 月份的湿度 (单位: %RH) 数据导出到文件 data.xlsx 中。部分数据如第 13 题图 a 所示。

统计 3 月 15 日每小时中湿度小于该日平均湿度的次数, 选择次数最多的前 5 个小时的数据按时间

(小时) 升序排序, 绘制如第 13 题图 b 所示的柱形图。

| 月 | 日 | 小时 | 分钟 | 湿度 |
|---|---|----|----|----|
| 3 | 1 | 0  | 0  | 78 |
| 3 | 1 | 0  | 1  | 73 |
| 3 | 1 | 0  | 2  | 75 |
| 3 | 1 | 0  | 3  | 74 |
| 3 | 1 | 0  | 4  | 72 |
| 3 | 1 | 0  | 5  | 72 |
| 3 | 1 | 0  | 6  | 73 |
| 3 | 1 | 0  | 7  | 71 |
| 3 | 1 | 0  | 8  | 75 |

第 13 题图 a



第 13 题图 b

实现上述功能的部分 Python 程序如下，请选择合适的代码填入划线处（单选）。

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_excel("data.xlsx")          # 读取文件
df1=df[df["日"]==15]                    # 筛选
① B
df1=df1[df1["湿度"]<ave]
② C
# 重命名 df2 中“湿度”列名称为“次数”，代码略
df2=df2.sort_values("次数", ascending=False) # 降序排序
df3=df2.head(5)                          # 获取前 5 条数据
```

③ **F**  
# 设置绘图参数，选取 df3 中的数据创建图表，显示如图 b 所示的柱形图，代码略  
程序中①②③处可选的代码有：

- A. ave=df["湿度"].mean() # 求均值
- B. ave=df1["湿度"].mean()
- C. df2=df1.groupby("小时", as\_index=False).count() # 分组计数
- D. df2=df1.groupby("湿度", as\_index=False).mean()
- E. df3=df2.sort\_values("小时", ascending=False)
- F. df3=df3.sort\_values("小时", ascending=True)

14. 在同一环境中加装另一光线传感器，现将 4 月份两个光线传感器每分钟的光照数据依次存储于列表 data 中。若两光线传感器相同时刻的数据差值超过阈值 M 的状态持续 N 分钟及以上，则将被判定为数据采集异常。现需找出 4 月份两个光线传感器数据采集异常的所有时间节点。请回答下列问题：

(1) 若 M 为 40, N 为 3, data 为[[402, 450], [301, 333], [375, 279], [447, 493], [607, 634]]，则 **▲ B**  
(单选：A. 存在 / B. 不存在) 数据采集异常。

(2) 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。

```
# 读入光照数据，按采集的时间顺序存储于列表 data 中，代码略
# 读入 M、N，代码略
res=[];flag=False;i=0
while i<len(data):
    d=data[i]
    if ① abs(d[0]-d[1])>M:
        if flag:
            ② cnt+=1
            if cnt>=N:
                # 根据 i 值计算时间节点并存入列表 res，代码略
            else:
                cnt=1
                flag=True
        else:
            ③ flag=False
    i+=1
```

15. 某仓库对物品实施入库与出库管理，每件物品对应唯一正整数编号。物品入库时，需将本次入库物品与库中已有物品重新整理，将编号连续的物品归为一堆，编号不连续的物品单独成堆，并根据各堆的物品数量升序排序。物品出库时，若当前待出库数量为 R，出库过程按以下步骤执行：

- ①全大于的情况：各堆的物品数量均大于 R，则从数量最小的堆中取出 R 个物品出库，过程结束。
- ②不大于的情况：若某堆的物品数量恰好等于 R，则将该堆全部出库，过程结束；否则从所有物品数量小于 R 的堆中，选取物品数量最大的一个（即最接近 R 的），将该堆物品全部出库，并更新 R ( $R = R - \text{该堆的物品数量}$ )，返回步骤①继续处理。请回答下列问题：

(1) 已知该仓库前 10 个入库的物品编号依次为 3, 6, 7, 1, 2, 11, 5, 10, 14, 13，且中途没有出库操作，则整理后仓库中的物品将分为 **▲ 4** (填数字) 堆。 **1-3, 5-7, 10-11, 13-14**

(2) 定义函数 proc(p, m)，用于实现将 m 指向的节点根据物品数量有序插入到链表 data 中。链表 data 中每个节点表示一个物品堆，由起始编号、终止编号、指针区域三项组成，已按各堆的物品数量升序排序。

```
def cnt(t): # 计算 t 指向的堆的物品数量
    return data[t][1]-data[t][0]+1
```

```
def proc(p,m):
    k=data[p][2]
    while k!=-1 and cnt(k)<cnt(m):
        p=k
        k=data[k][2]
    data[p][2]=m
    data[m][2]=k
```

将data[3]节点按每堆个数升序插入到head=2的data链表中

若 data 为[[1,1,4], [3,3,0], [8,8,1], [5,6,2], [10,15,-1]], 执行语句 proc(2,3)后, data[3][2]的值为 4 ▲。

(3) 模拟仓库处理入库和出库操作的 Python 程序如下, 请在划线处填入合适的代码。

```
def stockin(head, id):
```

# 将编号 id 的物品入库, 返回链表头指针 head, 代码略

```
def stockout(head, n): 在head开始的链表中, 取出n个
```

```
    res=[]
```

```
    while n!=0:
```

```
        if cnt(head)>n: 情况1: 全大于的情况
```

# 从 head 指向的堆取 n 个物品, 将其编号加入列表 res, 代码略

```
        data[head][0]+=n
```

```
        n=0
```

```
    else 情况2: 找最接近n或等于n的节点 else的条件是cnt(head)<=n
```

```
        q=-1
```

```
        p=head 从head的下一个节点开始找, 即data[p][2]
```

```
        while data[p][2]!=-1 and ① cnt(data[p][2])<=n
```

```
            q=p
```

```
            p=data[p][2]
```

```
        n=n-cnt(p)
```

# 取出 p 指向的堆中全部物品, 将其编号加入列表 res, 代码略

```
        if p==head:
```

```
            ② head=data[head][2]
```

```
        else:
```

```
            data[q][2]=data[p][2]
```

```
    return res, head
```

```
"""
```

获取仓库所有操作数据, 存储在列表 items 中, 每个元素包含两个数据项, items[i][0]为"in"时表示入库操作, 入库物品编号为 items[i][1]; items[i][0]为"out"时表示出库操作, 出库物品数量为 items[i][1], 代码略

```
"""
```

```
data=[];head=-1;total=0
```

```
for item in items:
```

```
    if item[0]=="in":
```

```
        head=stockin(head, item[1])
```

```
        total+=1 统计入库的数量
```

```
    else: 出库操作
```

```
        if total>=item[1]: 库内数量足够出库
```

```
            result, head=stockout(head, item[1])
```

```
            ③ total -= item[1] 或total -= len(result) 或total -= len(res)
```

# 按要求输出 result 的内容, 代码略

```
        else:
```

# 取消本次需求, 代码略