

高三上信息选考限时训练6 (作业39)

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、错选、多选均不得分。)

阅读下列材料, 回答第 1 至 2 题:

某医院 AI 辅助诊断系统, 可获取 CT 设备扫描得到的医学影像数据, 并分析生成有文字描述、带标注示意图及参数对比的诊断报告。医生审核后, 系统将诊断报告的查询链接发送至患者手机。

- 下列关于数据的说法, 正确的是 **D**
 - 系统中的医学影像是模拟信号 **是数字信号**
 - 诊断报告仅存储在患者手机中 **也储存在服务器中**
 - 诊断报告中的数据只有文字形式 **诊断报告带有文字描述、示意图等, 不仅仅只有文字**
 - 生成诊断报告的过程体现数据可加工处理
- 下列关于该系统中数据安全与保护的做法, 不合理的是 **C**
 - 定时备份系统数据
 - 加密存储患者诊断数据
 - 未经授权将所有就诊数据用于 AI 训练
 - 医生身份验证后才可查看患者诊断报告

阅读下列材料, 回答第 3 至 6 题:

某校配置的“智能助学系统”包括助学终端和服务器等。其助学终端含摄像头、触控屏等, 能获取语音、图片, 从中识别出题目信息上传至服务器; 服务器进行解题, 并将答案、解析等回传至助学终端供查看, 同时记录错题, 用于推送针对性学习资源。

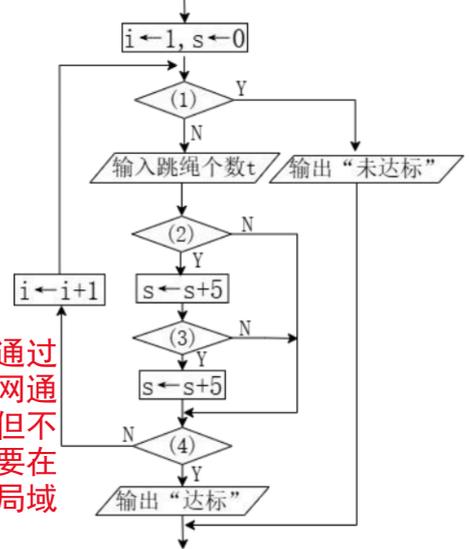
- 助学终端利用神经网络识别题目信息, 下列说法不正确的是 **C**
 - 识别题目应用了人工智能技术
 - 提高图片清晰度有助于提升识别精准度
 - 每次识别题目时, 需要重新学习原始的训练数据
 - 为解决语音识别准确率较低的问题, 可调整训练模型
- 下列关于该系统硬件和软件的说法, 不正确的是 **A**
 - 系统硬件是指助学终端和服务器
 - 触控屏既作为系统的输入设备也作为输出设备
 - 助学终端需要安装系统软件
 - 软件升级可为系统增加新功能

在该系统使用前原始数据就已经训练好, 识别时不需要重新训练, 否则效率极低

信息系统中的硬件是看得见摸得着的设备, 该系统中硬件除了助学终端和服务器外, 还有网络设备等等

- 下列关于该系统网络技术的说法, 不正确的是 **B**
 - 助学终端与服务器通信需遵守 TCP/IP 协议
 - 助学终端与服务器必须处于同一局域网内
 - 助学终端可通过有线网络传输数据
 - 一台服务器支持多台终端, 体现网络的资源共享
- 某张作业图片, 格式为未经压缩的 BMP, 分辨率为 1920×1080 像素, 量化位数为 24 位, 则该图片的存储容量约为 **B**
 - 2MB
 - 6MB
 - 18MB
 - 56MB

可以通过互联网通信, 但不一定要在同一局域网内



第 7 题图

$$1920 \times 1080 \times 24 / 8 / 1024 / 1024 = 6MB^{89}$$

7. 某校组织“天天跳绳”活动，1周跳绳总积分(s)累计达到35分及以上为“达标”，否则“未达标”。每天跳绳(t)大于160个计10分，大于50个少于160个计5分，少于50个计0分。判断是否达标的部分流程图如第7题图所示，(1)~(4)处可选表达式为：① $s \geq 35$? ② $t \geq 50$? ③ $t \geq 160$? ④ $i > 7$?

则(1)~(4)处表达式序号依次为 C

- A. ④③②① B. ①②③④ C. ④②③① D. ①③②④

8. 某栈A中，栈底到栈顶的元素依次为5, 1, 4, 3, 2，栈B初始为空。栈A元素出栈后可直接输出或进入栈B，栈B元素出栈后也可直接输出或进入栈A。若输出次序为1, 2, 3, 4, 5，则入栈次数至少为 B

- A. 3 B. 5 C. 6 D. 8

9. 某二叉树的中序遍历是BAC,则其前序遍历不可能是 C

- A. ABC B. CAB C. ACB D. BAC

10. 某Python程序段如下:

```
i=0; j=len(a)-1; key=int(input())
```

```
while a[i]<=key<=a[j]:
```

```
    m=(i+j)//2
```

```
    if a[m]==key:
```

```
        break
```

```
    elif a[m]>key:
```

```
        j=m-1
```

```
    else:
```

```
        i=m+1
```

循环退出有两种情况:

在循环内 $a[m]==key$ 时 break, 此时

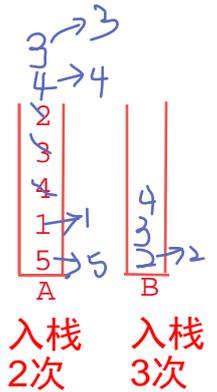
i, j 是进入该轮循环时的值, m 在 $[i, j]$ 之间。

循环条件 $a[i] \leq key \leq a[j]$ 不成立时退出。

二叉树最后一层叶子节点查找, $a[i]$ 和 $a[j]$

是同一个值, 若该值不是要找的值, 无法进入

循环, 所以, i 不会出现大于 j 的情况



若 a 为 [1, 3, 5, 7], 运行该程序段后, i 和 j 的值不可能是 A

- A. 3, 2 B. 2, 3 C. 0, 3 D. 0, 0

11. map 和 ker 均为二维数组, 分别存储如第 11 题图 a 和第 11 题图 b 所示的数据。

```
def ct(x, y, m):
```

```
    s=0
```

```
    for i in range(x, x+m):
```

```
        for j in range(y, y+m):
```

```
            s+=map[i][j]*ker[i-x][j-y]
```

```
    return s
```

```
map=[[3, 247, 4], [2, 180, 3], [2, 101, 1]]
```

```
ker=[[-1, 1], [-1, 1]]
```

```
n=len(map); m=len(ker)
```

```
c=[]
```

```
for i in range(n-m+1):  $n=3, m=2, i, j$  范围是  $[0, 1]$ 
```

```
    for j in range(n-m+1):
```

```
        c.append(ct(i, j, m))
```

```
print(c)
```

执行该程序段, 输出的结果是 B

- A. [-420, 422, -277, 277] B. [422, -420, 277, -277]
C. [422, 277, -420, -277] D. [-277, 277, -420, 422]



第 11 题图 a

第 11 题图 b

12. 用列表 d 模拟的链表结构如图 a 所示, 该链表的数据存储规则为: 所有负数都保存于链表前部。要求实现功能: 输入 x 值 ($x > 1$), 如果链表中第 x 个节点 (头节点是第 1 个节点) 的数据区域中的数值为负数, 则将其值取反, 并通过改变链接关系, 使该节点放在链表尾部, 数据保存依旧符合存储规则。设 x 为 2, 该链表调整后结果如图 b 所示。

#部分代码如下:

```
q = h
for i in range(1, x):
    p = q
    q = d[q][1]
if d[q][0] < 0:
    (1)
    (2)
```

```
while d[p][1] != -1:
    p = d[p][1]
(3)
(4)
```

则 (1) 至 (4) 处的语句依次为 C

①d[p][1] = q ②d[q][1] = -1 ③d[p][1] = d[q][1] ④d[q][0] = -d[q][0]

- A. ①②③④ B. ②①④③ C. ③④①② D. ①③②④

	数据区域	指针区域
0	-7	3
1	-9	0
2	8	-1
3	-1	5
4	5	2
5	4	4

图a

	数据区域	指针区域
0	7	-1
1	-9	3
2	8	0
3	-1	5
4	5	2
5	4	4

图b

定位到x节点, q在x节点上

q节点为负数, 则变正数, 并从当前位置删除

p走到尾节点

q加在p之后

两句可以交换

二、非选择题 (本大题共3小题, 其中第13小题7分, 第14小题10分, 第15小题9分, 共26分)

13. 某电商平台每分钟监测当前排队等待客服服务的用户数, 根据等待用户数动态调整客服模式 (人工模式或“AI+人工”混合模式)。当排队用户数超过上限阈值 N 时, 系统切换到混合模式; 当排队用户数低于下限阈值 M ($M < N$) 时, 系统切换到人工模式。每种模式一旦激活, 至少持续 3 分钟。系统记录每次混合模式的起止时间点, 为招聘客服提供数据支持。

(1) 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。

```
#设置上限阈值 N 和下限阈值 M, 代码略
c=0
ct=[] #记录混合模式的起止时间点
mode=0 #0 表示人工模式, 1 表示“AI+人工”混合模式
ctime=1
while True:
    #读取当前排队用户数, 存入 s, 代码略
    if s>N and mode==0 or (1):
        s<M and mode==1或
        s<M and mode!=0
        c=c+1
        if (2)=3 或 c>=3:
            mode=1-mode
            ct.append(ctime)
            c=0
    else:
        c=0
        (3) ctime+=1
        #延时 1 分钟, 代码略
```

(2) 下面代码实现的功能是输出 ct 中每次混合模式的起止时间, 加框处代码有误, 请改正。

```
i=0
while [i<len(ct)]: i<len(ct)-1
    st=ct[i]
    ed=ct[i+1]
    #输出混合模式的起止时间点, 代码略
    i=i+2
```


用列表 d 模拟的链表结，该链表的数据存储规则为：链表中所有数值升序。要求实现功能：如果链表中节点中数据区域中的数值为负数，则将其值取反，并通过改变链接关系，使该节点放在链中合适位置，数据保存依旧符合存储规则。

```
p = -1; q = head
```

```
while q != -1:
```

```
    r = d[q][1]
```

```
    if d[q][0] < 0:
```

```
        d[q][0] = -d[q][0]
```

```
        c = d[q][0]
```



```
insert_p = -1
```

```
insert_q = head
```

```
while insert_q != -1 and d[insert_q][0] <= c:
```



```
if insert_p == -1:
```

```
    d[q][1] = head
```

```
    head = q
```

```
else:
```



```
else:
```



```

p = -1; q = head
while q != -1:
    r = d[q][1]
    if d[q][0] < 0:
        d[q][0] = -d[q][0]
        c = d[q][0]
        if p == -1:
            head = r
        else:
            d[p][1] = r
    insert_p = -1
    insert_q = head
    while insert_q != -1 and d[insert_q][0] <= c:
        insert_p = insert_q
        insert_q = d[insert_q][1]
    if insert_p == -1:
        d[q][1] = head
        head = q
    else:
        d[insert_p][1] = q
        d[q][1] = insert_q
else:
    p = q
    q = r

```