湖州、衢州、丽水 2024 年 11 月三地市教学质量检测试卷

高三技术

本试题卷分两部分，第一部分信息技术，第二部分通用技术。全卷共 12 页，第一部分 1 至 6 页，第二部分 7 至 12 页。满分 100 分，考试时间 90 分钟。

1. 考生答题前，务必将自己的姓名、准考证号用黑色字迹的签字笔或钢笔填写在答题纸上。
2. 选择题的答案须用 2B 铅笔将答题纸上对应题目的答案标号涂黑，如要改动，须将原填涂处用橡皮擦净。
3. 非选择题的答案须用黑色字迹的签字笔或钢笔写在答题纸上相应区域内，作图时可先使用 2B 铅笔，确定后须用黑色字迹的签字笔或钢笔描黑，答案写在本试题卷上无效。

第一部分　信息技术（50 分）

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 3 题：

某智慧校园系统集成了师生基本信息、教学活动记录以及系统设备状态等多类型数据，这些数据涵盖了文本、图像、音视频等多种形式。该系统借助人工智能技术分析服务器中的数据生成个性化学习建议，师生可通过终端设备进行查阅。

1. 关于智慧校园中数据的叙述，正确的是 B

A. 文本数据的价值大于音视频数据
B. 数据以二进制的形式存储在计算机中
C. 音视频数据容量大，是大数据
D. 该系统中的数据均为非结构化数据

2. 下列对于师生教学活动数据的处理方式，不合理的是 D

A. 定期删除重复的数据
B. 对教学视频的语音进行识别并生成字幕
C. 为了方便传输，对视频文件进行压缩
D. 为了节省存储空间，降低声音文件的音量

3. 下列措施中，不能有效地改善系统生成个性化建议的速度和精准度的是 B

A. 升级服务器的硬件配置
B. 提升终端设备的性能　提升服务器的性能
C. 提高数据质量
D. 提高生成个性化建议算法的效率

阅读下列材料，回答第 4 至 6 题：

某餐厅点餐系统有如下功能：顾客通过手机扫码点餐，订单数据保存到服务器的数据库中，系统再将数据传输到厨师、服务员和收银员使用的智能终端设备上，顾客可以在手机上查看餐品的制作进度。

4. 下列关于该系统组成与功能的说法，正确的是 A

A. 收银员使用的智能终端设备属于硬件
B. 智能终端设备无须安装系统软件
C. 数据库中仅保存了订单数据
D. 该系统不具备数据输出功能

5. 下列关于该系统安全的做法，合理的是 C

A. 随意共享客户信息
B. 以明文方式存储账号密码
C. 定期备份数据
D. 为多个收银员设置同一个账号

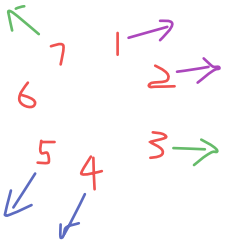
6. 下列关于该系统中网络的说法，正确的是 C

A. 服务器和网络互联设备构成该餐厅的网络

B. 服务器网络故障不影响顾客使用手机下单

C. 终端设备与服务器之间的通信需要遵循网络协议

D. 手机只有通过 Wi-Fi 接入餐厅局域网才能查看制作进度



7. 有 1 个队列，队首到队尾的元素依次为 1,2,3,4,5,6,7。现进行如下操作：①出队 2 个元素；②再出队 1 个元素并将该元素入队。重复以上操作，直至队列中仅剩 1 个元素。则该元素是 C

A. 3　　B. 5　　C. 6　　D. 7

8. 某二叉树的后序遍历序列为 ABCDE，其中节点 A 和节点 B 分别是节点 C 的左右孩子，则该树的前序遍历可能是 D

根

A. CABED　　B. DCABE　　C. EACBD　　D. EDCAB

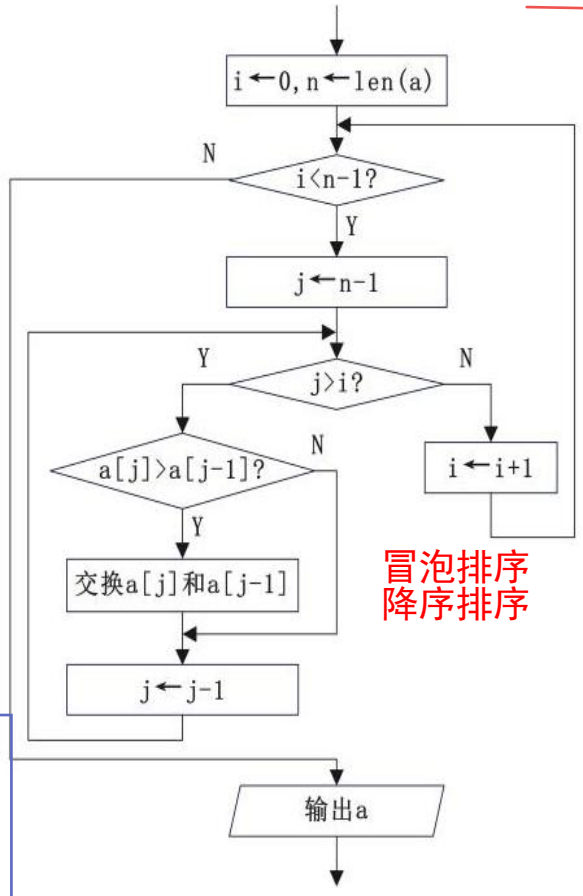
E
D
C
CAB
A B

9. 某算法的部分流程图如图所示，若数组 a 初始值依次为 5,3,2,4,7，执行该部分流程后，输出数组 a 的值为 B

降序

A. 2,3,4,5,7　　B. 7,5,4,3,2

C. 2,5,7,4,3　　D. 7,4,3,5,2



冒泡排序
降序排序

第 9 题图

10. 定义如下函数：

```
def f(x,y,flag):
    if flag == False:
        return x*y//f(x,y,not flag)
    if x == y:
        return x
    elif x < y:
        return f(x,y-x,flag)
    else:
        return f(x-y,y,flag)
```

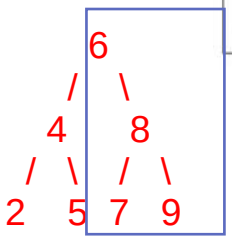
f(6,9,flag)
=6*9//f(6,9,True)
=54//f(6,3,True)
=54//f(3,3,True)
=54//3
=18

执行语句 w = f(6,9,False)后，w 的值为 A

A. 18　　B. 9　　C. 6　　D. 3

11. 某二分查找算法的 Python 程序段如下：

```
key,cnt = 5,0
i,j = 0,len(d)-1
while i <= j:
    cnt += 1
    m = (i+j)//2
    if key == d[m]:
        break
    elif key < d[m]:
        j = m-1
    else:
        i = m+1
```



在右边加入元素，不影响key的查找次数

当 d 为[2,4,5,6,7,8,9]时，在 d 中插入一个元素 x，维持 d 为一个非降序序列，在元素 x 插入前后分别运行程序段，若变量 cnt 的值相同，则增加的元素 x 可能是 D

A. 2　　B. 3　　C. 4　　D. 6

12. 使用列表 d 模拟链表结构（节点数大于 0），每个节点包含数据区域和指针区域，h 为头指针（头节点数据为奇数），如图 a 所示。现需修改该链表各节点的链接关系，使链表各节点数据区域中的数值按奇数在前，偶数在后排列，结果如图 b 所示。实现该功能的程序段如下，方框中应填入的正确代码为 A

```
p = h
q = d[h][1]
r,tmp = -1,-1
while q != -1 :
    if d[q][0]%2 == 0:
        [          ]
    else:
        d[p][1] = q
        p = q
    q = d[q][1]
if d[p][1] == -1 and r != -1:
    d[r][1] = -1
d[p][1] = tmp
```

q为偶数
若原来偶数链空
tmp为q
若不为空
q插入在r之后
不管链表是否为空
链尾r都要更新到q上

7 1 6 ..

	数据区域	指针区域
0	5	-1
1	2	0
2	3	1
h→ 3	7	5
4	4	2
5	1	6
6	6	4

第 12 题图 a

7 1 3 5 6..

	数据区域	指针区域
0	5	6
1	2	-1
2	3	0
h→ 3	7	5
4	4	1
5	1	2
6	6	4

第 12 题图 b

r是偶数的链尾，tmp是偶数的链头
p是奇数的链尾

A.
```
if r == -1:
    tmp = q
else:
    d[r][1] = q
r = q
```

B.
```
if r == -1:
    tmp = q
else:
    d[r][1] = q
    r = q
```

C.
```
if r == -1:
    tmp = q
    r = q
    d[r][1] = q
r = q
```

D.
```
if r != -1:
    d[r][1] = q
else:
    tmp = q
    r = q
```

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. 某公司举办抽奖活动，设有一、二、三等奖。每张奖券印有 3 个互不相同的数字，每次在兑奖前公布中奖号码，兑奖时不考虑中奖数字的先后顺序，兑奖规则如下表所示。编程统计奖券获得的总奖金。

奖项	一等奖	二等奖	三等奖
兑奖条件	3 个数字均匹配	2 个数字匹配	1 个数字匹配
奖金	500 元	200 元	100 元

（1）某次中奖号码为 3,1,8，则号码为 1,6,3 的奖券可以获得奖金 ▲ 元。 200

（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。

```
# 输入中奖号码，依次存储到列表元素 win[0]至 win[2]中，代码略
rew = [0,100,200,500]
mon=0 ①
n = int(input("输入奖券数量："))
for i in range(n) ② :    遍历所有的奖券
    cnt = 0
    # 输入奖券的号码，依次存储到列表元素 tic[0]至 tic[2]中，代码略。
    for j in range(3):
        if tic[j] in win:
            cnt += 1
    mon = mon+rew[cnt] ③    累加
print("总奖金金额为：",mon)
```

多选题的判断思路是否相同？

14. 某小组拟搭建一个系统进行水质监测，采集鱼塘的水温、溶解氧以及 pH 值等数据。该系统有若干个监测点，每个监测点均配备智能终端、传感器、执行器和 IoT 模块，智能终端通过 IoT 模块连接 Web 服务器上传水温等数据，并从服务器获取阈值，若氧浓度低于阈值，系统将启动增氧机；若水质异常，系统将异常信息发送至用户手机。用户可以通过浏览器查看水质数据。请回答以下问题：

（1）下列关于该信息系统中数据传输的说法，正确的是 ▲A（单选，填字母）

A. 智能终端与 Web 服务器的数据传输是双向的

B. 数据在 Web 服务器与浏览器之间的传输是单向的

C. 不同监测点的数据必须传输到不同的 Web 服务器

（2）下列关于该信息系统支撑技术的说法，正确的是 ▲BCD（多选，填字母）

A. 不同监测点的传感器型号必须相同

B. 溶解氧的阈值可以存储在智能终端

C. 实时判断水质是否异常的程序可以在智能终端运行

D. 若监测点的 IoT 模块损坏，可以使用有线的方式将智能终端连接到 Web 服务器？

（3）监测点 A 的 pH 值传感器编号 tid 为 2，获取的 pH 值 t 为 6.8，提交数据到 Web 服务器的 URL 为 http://192.168.1.6:5000/toph?tid=2&t=6.8，则 Web 服务器接收数据的路由是 ▲/toph。

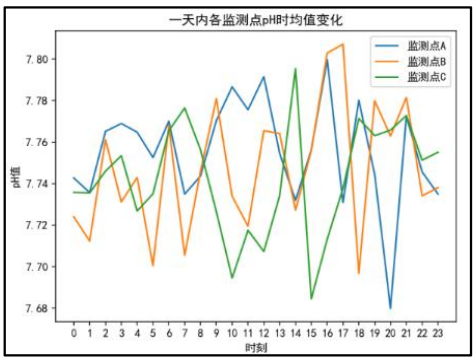
数据原因：阈值设置不合理（过高）；采集到的数据一直低于阈值；
软件原因：缺乏关闭增氧机的程序；

（4）系统运行一段时间后，发现每个监测点的增氧机都一直在工作，请从软件或数据的角度描述出现该现象的可能原因。（回答 2 项，1 项正确得 1 分）

（5）导出某天的数据，整理后的部分数据如图 a 所示，分析每个监测点每小时的 pH 平均值，可视化后的结果如图 b 所示，请在划线处填入合适的代码。

监测点	时	分	秒	水温	pH	溶氧量
监测点A	0	59	57	25.7	7.4	6.13
监测点B	0	59	57	25.7	7.2	5.79
监测点C	0	59	57	24.5	6.5	4.86
监测点A	1	0	0	25.7	7.5	6.21
监测点B	1	0	0	25.7	7.5	5.5
监测点C	1	0	0	24.5	6.5	5.1
监测点A	1	0	3	25.7	7.5	6.25
监测点B	1	0	3	25.7	7.5	5.6
监测点C	1	0	3	24.5	6.5	5.1

第 14 题图 a



第 14 题图 b

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("water_data.csv")
for pos in "ABC":
    sta = "监测点" + pos
    dftmp = df[_____①_____]
    gtmp = dftmp.groupby("时", as_index=False)["pH"].mean()
    plt.plot(_____②_____, gtmp.pH)
# 设置绘图参数，显示如图 b 所示的线型图，代码略
```

df.监测点==sta 或者 df["监测点"]==sta

gtmp.时 或者gtmp["时"]

15. 某区域有 m 个设备（编号为 1 到 m）需要联网。单台无线路由器的 Wi-Fi 信号可以覆盖一个圆形范围。现有 n 个可供安装路由器的位置（编号为 0 到 n-1）。编写程序，找到最少数量的路由器安装位置，确保所有设备都能被覆盖。

查找算法如下：　这种算法是什么算法？是否是全局最优解？

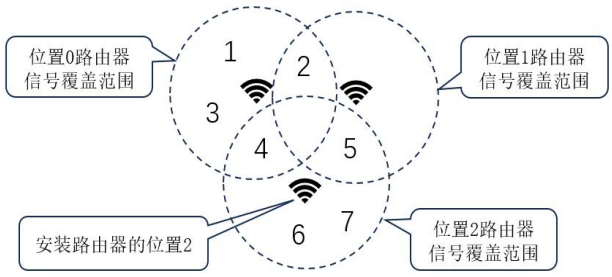
步骤 1：选择一个覆盖设备最多的位置安装路由器。在设备数量相同的情况下，优先选择编号较小的安装位置。

步骤 2：将该路由器覆盖范围内的设备，从其他安装位置的覆盖范围中移除。

步骤 3：在剩余安装位置中继续选择覆盖设备最多的安装位置并安装路由器。

步骤 4：重复上述步骤，直到所有设备都被覆盖。

若 m 的值为 7，n 的值为 3，每个安装位置的路由器能覆盖的设备如图所示。位置 0 的路由器能覆盖编号为 1 至 4 的设备，设备编号区间用列表[1,4]表示。在位置 2 的覆盖设备中移除设备 4，位置 2 覆盖设备编号区间为[5,7]，2 台路由器的覆盖区间为[1,7]，能覆盖所有设备。



第 15 题图

程序中用到的列表函数与方法如下表所示：

函数与方法	功能
lst.append(x)	在列表 lst 末尾添加元素 x
lst.insert(i,x)	在列表 lst 中下标为 i 位置处插入元素 x
lst.pop(i)	将列表 lst 中下标为 i 的元素删除

（1）若有 4 个位置可以安装路由器，每个安装位置覆盖的设备编号如下表所示，需覆盖 7 个设备，按上述查找算法，则安装路由器的位置依次是__▲0,3__。

安装位置	0	1	2	3
设备编号	1,2,4,7	1,5,7	2,3,4,6	3,5,6,7

（2）定义如下 plan(left,right,segs)函数，参数 left 和 right 分别表示一个连续设备编号区间的开始和结束位置，参数 segs 存储所有已经覆盖的设备编号区间。若 segs 列表不空，则各区间升序排列，如[[8,9],[11,13],[15,15]]。程序的功能是将区间[left,right]合并到列表 segs 中，并保持各个区间升序排列。

```
def plan(left,right,segs):
    if len(segs) == 0 or right+1 < segs[0][0]:
        segs.insert(0,[left,right])
    # 处理 rihgt 位于区间 segs[0]右侧的情况，代码略。
```

若函数中加框处语句误写为“right < segs[0][0]”，会导致某些情况下无法得到符合函数功能的结果。调用 plan(left,right,segs)函数，若 left 为 3，right 为 5，能测试出这一问题的 segs 值为__▲BD__（多选，填字母）。

A. [[8,9],[11,13]]　　B. [ ]　可测试出缺少len(segs)==0

C. [[1,2],[7,9]]　　D. [[6,7],[9,10]]

[3,5] 5+1=6

if 判断规则：　若segs=[]

len(segs) == 0 or right+1 < segs[0][0]

True　　?

（3）实现查找最少安装位置功能的部分 Python 程序如下，请在划线处填入合适代码。

```
def search(lst):
    # 函数功能：返回列表 lst 中长度最大元素的下标，若有多个，返回第 1 个，代码略。
def dellst(x,sta,lst):
    '''
    参数 x 为设备编号，参数 sta 存储各个设备可被覆盖的安装位置，如元素
    sta[1]值为[0,2]，表示能覆盖 1 号设备的路由器可以安装在编号为 0 和 2
    的两个位置。函数功能：删除列表 lst 中值为 x 的数据项。
    '''
    for d in sta[ X ①  ]:
        i = 0
        while lst[d][i] != x:
            i += 1
        lst[d].pop(i)
'''
```

实际就是lst[?].remove(d)

读取需覆盖的设备数量存入 m，可供安装路由器的位置数量存入 n；
读取各个安装位置能覆盖设备的编号，依次存入列表 lst 中；
如 lst=[[1,5,7],[2,3,4,6],[3,5,6,7]]，元素 lst[0]表示安装位置 0 可以覆盖编号为 1,5,7 的设备。列表 lst 各元素的数据项升序排列。代码略。

```
'''
sta = [[]for i in range(m+1)]
for i in range(len(lst)):
    for d in lst[i]:
        ② sta[d].append(i)
ans,segs = [],[]
while segs != [[1,m]]:
    pos = search(lst) #返回列表 lst 中长度最大元素的下标，若有多个，返回第 1 个
    tmp = [d for d in lst[pos]] # 将列表 lst[pos]各个元素复制到 tmp 列表
    i,j = 0,0
    while j < len(tmp):
        while j < len(tmp)-1 and ③ :
            j += 1
        plan(tmp[i],tmp[j],segs)
        j = j+1
        i = j
    ans.append(pos)
    for x in tmp:
        dellst(x, sta, lst)
# 依次输出安装路由器的位置及数量，代码略。
```

sta变量和设备有关

统计各个设备可被覆盖的安装位置

lst[i]为第i位置的路由器能覆盖所有设备清单，sta[d]存储能覆盖i号设备的路由器可以安装的位置清单

#存储结果位置和当前已覆盖的设备区间

#遍历所有设备

tmp[j]+1==tmp[j+1]

#若找到连续的设备号一直往后找，直到设备号不连续，tmp[i],tmp[j]为开始和结束设备号

区间合并

[1,2,4,7]

lst 列表： 路由器 能连接的设备

路由器位置	0	1	2	3
设备编号	1,2,4,7	1,5,7	2,3,4,6	3,5,6,7

反向数据

sta 列表： 设备能连的路由器

设备编号	0	1	2	3	4	5	6	7
路由器		0,1	0,2	2,3	0,2	1,3	2,3	0,1,3

建立反查列表sta的优点：如果数据量很大，可以快速定位，不需要在lst中穷举

解法2：不添加额外辅助列表，直接在lst上操作 删除时，直接遍历lst，不通过sta辅助列表

```
lst = [[1, 2, 4, 7], [1, 5, 7], [2, 3, 4, 6], [3, 5, 6, 7]]
m = 7  # 需要连入的设备
n = 4  # 路由器位置
cnt = 0
while cnt < m:
    st = 0
    while len(lst[st]) == 0:  # 找到非空 第1个路由器编号
        st += 1
    max_index = st
    for i in range(st+1, len(lst)):
        if len(lst[i]) > len(lst[max_index]):
            max_index = i
    print("找到路由器编号", max_index)
    cnt += len(lst[max_index])
    for i in range(len(lst)):  # 确定一个路由器，从其他路由器中删除编号，自己变为[]
        if i != max_index and len(lst[i]) > 0:
            tmp = []
            for x in lst[i]:  # x
                if x not in lst[max_index]:
                    tmp.append(x) #不删除的留下来
            lst[i] = tmp #留下来的重新赋值给lst[i]
    lst[max_index] = []  # 已经覆盖，就清空
```