

高三上信息选考练习卷11

第一部分 信息技术（共 50 分）

一、选择题（本大题共 12 小题，每小题 2 分，共 24 分。在每小题给出的四个选项中，只有一个符合题目要求）

阅读下列材料一，回答第 1 至 3 题：

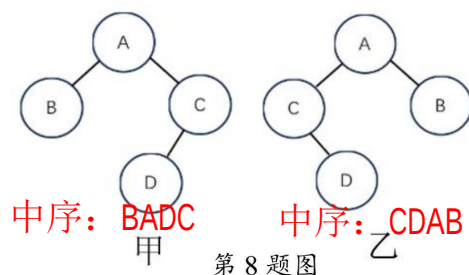
某校学生可通过移动终端提前预订餐品并完成支付，食堂则根据预订数据进行精准备餐。智慧食堂系统后台能够实时监控各菜品的销售情况，并生成每日、每周的销售报表与营养分析报告，为食堂优化菜单、合理采购提供数据支持。

1. 关于该系统中的数据，下列说法正确的是 D
 - A. 学生提交订单的历史记录不属于数据
 - B. 学生完成支付的过程中不会产生新的数据
 - C. 记录在系统中的销售报表，属于模拟数据
 - D. 通过消费数据来调整菜单，体现了数据的应用价值
2. 下列措施中，不能有效保障智慧食堂系统的数据安全的是 D
 - A. 对学生用户的密码加密后存储
 - B. 定期对员工进行系统操作和网络安全培训
 - C. 部署网络防火墙，防止未经授权的外部访问
 - D. 在菜品加工区安装高清摄像头
3. 学生通过移动终端查看菜品图片、价格并完成在线预订。该过程主要涉及的技术不包括 C
 - A. 数据库技术
 - B. 移动通信技术
 - C. 全球定位系统
 - D. 人机界面设计

阅读以下材料二，完成第 4 至 7 题：

某现代农业示范园部署了一套智能温室监控系统。该系统通过多种传感器实时采集环境数据，数据通过物联网传输至云端服务器进行存储和深度分析。系统内置的 AI 模型能够根据作物生长图像，自动识别病虫害并发出预警。管理员可以通过电脑或手机 APP 远程查看实时数据、接收预警信息，并控制通风、灌溉、补光等设备。

4. 有关该智能温室监控系统的描述，正确的是 **A**
- A. 系统使用的传感器中一定包括图像传感器
 - B. 数据不必经过数字化也可在系统中存储和传输
 - C. 管理员远程控制通风设备时不需要网络的支持
 - D. 该系统一旦开发完毕，可以完全脱离操作系统而单独运行
5. 关于该系统服务控制功能的设计方案，不合理的是 **D**
- A. 当温度传感器采集的温度值高于预设值时则启动通风机降温
 - B. 当光敏传感器采集的亮度值低于预设值时就启动补光灯
 - C. 当湿度传感器采集的湿度值低于预设值时则打开水泵灌溉
 - D. 当预警信息发出后就清空数据库以节省存储空间
6. 该系统利用深度学习模型识别作物病虫害，该模型的识别准确性主要依赖于 **B**
- A. 事先手工构造的知识库 **符号主义**
 - B. 事先标注好的训练数据集 **联结主义**
 - C. 病虫害的精准文字解释 **符号主义**
 - D. 防治作物病虫害的专家系统 **符号主义**
7. 综合材料一“智慧食堂”和材料二“智能温室”两个系统，最能体现它们的共通之处是 **A**
- A. 通过分析历史或实时数据，为优化管理提供决策支持
 - B. 依据传感器采集的数据，对物理设备进行实时的自动控制
 - C. 经由专用 APP 提供人机交互功能，都基于 B/S 架构开发 **C/S**
 - D. 将海量的结构化与非结构化数据，集中存储于云端服务器
8. 将原二叉树中所有节点的左右子树都进行互换，得到的新二叉树，称之为原二叉树的镜像二叉树。如第 8 题图所示，甲、乙二叉树互为镜像二叉树。已知某二叉树的中序遍历序列为 B-D-A-C-E，则镜像二叉树的中序遍历序列为 **C**
- A. C-E-A-B-D
 - B. C-E-A-D-B
 - C. E-C-A-D-B
 - D. E-C-A-B-D



While 队列非空:
 If 栈非空, 且栈顶元素大于队首元素 then
 将栈顶元素出栈后入队
 Else 若栈顶元素小于队首
 将队首元素出队后入栈

大小

第 9 题图

- A. 14, 7, 5, 20, 9, 16 B. 5, 20, 7, 16, 9, 14 C. 5, 7, 9, 14, 16, 20 D. 20, 16, 14, 9, 7, 5

10. 有如下 Python 程序段:

#suma 存放字符串 st 所有字符的 ASCII 码之和

```
sumL = 0; sumR = suma; pos = -1
```

```
for i in range(0, len(st)):
```

```
    sumR = sumR - ord(st[i])
```

```
    if sumR == sumL:
```

```
        pos = i
```

```
    sumL = sumL + ord(st[i])
```

字符串 st 为"dbhacdbahcd", 执行该程序段后, pos 的值为 B

A. -1

B. 5

C. 6

D. 7

11. 甲乙两名玩家按如下规则轮流进行游戏: 桌上有 n 枚硬币, 玩家每次可以从中拿走 1 枚或者 2 枚。谁拿走最后一枚硬币谁就输掉了游戏。为求出胜者, 定义如下函数:

```
def g(n, cur):
```

```
    if n <= 0:
```

```
        return cur
```

```
    if cur == "甲":
```

```
        nxt = "乙"
```

```
    elif cur == "乙":
```

```
        nxt = "甲"
```

```
    if g(n - 1, nxt) == cur or g(n - 2, nxt) == cur:
```

```
        return cur
```

```
    else:
```

```
        return nxt
```

当n=1, cur=甲时, 甲一定会输

当n=2, cur=甲时, 甲可以通过只拿1个,

即g(n-1,nxt)-cur 获得胜利

当n=3, cur=甲时, 甲可以通过拿2个,

即g(n-2,nxt)=cur 获得胜利

当n=4, cur=甲时, 甲无论拿1个还是拿2个,

n=2或n=3的必胜条件都在乙手上, 故甲必输

当n=5, cur=甲时, 甲可以通过拿1个,

制造n=4的必输条件给乙, 故甲必赢

当n=6, cur=甲时, 甲可以通过拿2个,

制造n=4的必输条件给乙, 故甲必赢

只有当n=4时, 甲无法胜利

执行如下语句后, s 的值与其他三个选项不同的是 B

A. s = g(3,"甲")

B. s = g(4,"甲")

C. s = g(5,"甲")

D. s = g(6,"甲")

12. 有如下 Python 程序段:

```
m, n = 0, data[p][0]
```

```
cur = p; tag = [p, p]
```

```
while p != -1:
```

```
    num = data[p][0]
```

```
    if m <= 0:
```

```
        m = num
```

```
        cur = p
```

```
    else:
```

```
        m += num
```

```
    if n < m:
```

```
        n = m
```

```
        tag[0], tag[1] = cur, p
```

```
    p = data[p][1]
```

该程序段实现的功能是:通过链表遍历, 维护当前子数组和m、最大子数组和n, 并记录最大和对应的起始(tag[0])和结束(tag[1])位置。

链表依次是:3->-4->4->-3->5->-9,

↑ ↑

最大子数组是 4(下标为 3)、-3、5(下标为4)

若 p 为 2, data 为[[-4,3],[-9,-1],[3,0],[4,5],[5,1],[-3,4]], 执行该程序段后, tag 的值为 B

A. [3,3]

B. [3,4]

C. [3,5]

D. [4,5]

二、非选择题（本大题共 3 小题，其中第 13 小题 8 分，第 14 小题 8 分，第 15 小题 10 分，共 26 分）

13. 某海啸预警中心在某海域布设了能监测海波高度的浮标。浮标中内置传感器和智能终端，智能终端每 15 秒从传感器获取一次海波高度数据，初步处理后，每分钟将数据经卫星上传至中心服务器。服务器实时计算海啸发生概率，并在概率超过阈值时向沿海地区发布不同等级的警报。公众可通过官方 App 或网站查看实时数据和预警信息。请回答下列问题：

- (1) 浮标中的智能终端 A (单选，选字母：A. 有 / B. 没有) 数据存储功能。
- (2) 该系统的部分数据处理在智能终端完成，可以 B (单选，填字母)。
- A. 提升中心服务器硬件的运算速度
- B. 降低数据传输的带宽和功耗
- C. 提高海波高度数据的采集精度
- (3) 若预警系统存在过多错误的报警，下列改进措施中最合理的是 C (单选，填字母)。
- A. 投放更多浮标，并人工复核所有报警数据
- B. 降低采样频率到每小时一次以减少波动 降低准确性
- C. 融合地震波数据，引入大数据预测模型

(4) 为了提高系统的预警准确率并减少漏报，还有什么改进措施？请列举一项，并说明如何改进。

增加单个浮标的传感器类型，进行多维度监测；

分析大量的历史数据或模拟海啸；以动态设置合适的阈值

(5) 若某采样时刻的海波高度大于其前后相邻采样时刻的高度，则该采样点为一个波峰；若存在连续若干采样点海波高度相同，且其高度均高于两侧采样点的高度，则该连续区间视为一个波峰。如在某一时间区间内采集到的海波高度数据 data 为 [1.72, 1.85, 2.01, 2.12, 2.12, 2.12, 1.92, 1.71, 1.32, 1.26, 1.33, 1.73, 1.92, 1.71, 1.32]，其中波峰的数量为 2。统计波峰数量的 Python 程序如下：

```
peaks = []
n = len(data)
    ▲ i=1
while i < n - 1:
    if data[i] > data[i-1]:
        j = i
        while j < n-1 and data[j] == data[j+1]:
            j += 1
        if data[j] > data[j+1]: j < n - 1 and data[j] > data[j+1]
            peaks.append([(i+j)//2, data[j]]) # 追加到列表 peaks 的末尾
        i = j
    i += 1
print("波峰个数:", len(peaks), ", 波峰位置和高度:", peaks)
```

while 循环内层先找升序区间，假设 i 是连续波峰的起点，j 用于找到连续波峰的终点，当 j 循环退出后，对应的条件应为 $\text{data}[j] > \text{data}[j+1]$ 或 $j = n-1$ ，故加框处应改为 $j < n-1 \text{ and } \text{data}[j] > \text{data}[j+1]$ 。
(peaks 记录波峰位置，若 j 循环退出时 $j = n-1$ ，说明之后没有下降区间，这一段也就不能称为波峰了)

前后顺序不能交换

①请在划线处补充代码。

②方框处代码有误，请改正。

14. 预警中心将某海域一年内采集到的海波高度数据存储在 wave.xlsx 文件中，部分数据如第 14 题图所示。现要对这些数据进行分析，找出海波高度异常（大于 1.5 米）记录最多的浮标，并找出该浮标异常次数最多的月份。

月	日	时间	浮标ID	海浪高度(m)	经度	纬度
1	1	00:00	0	1.7	122.35	30.05
1	1	00:00	1	1.4	121.48	28.62
1	1	00:00	2	1.9	120.83	27.85
1	1	00:01	0	1.5	122.35	30.05
1	1	00:01	1	1.4	121.48	28.62
1	1	00:01	2	1.9	120.83	27.85

第 14 题图

- (1) 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("wave.xlsx")      # 读取 Excel 文件
df_anomaly = df[df["海波高度(m)"] > 1.5]
df_eare = ① A 通过对“浮标ID”进行分类汇总 count 计算次数
df_eares = df_eare.sort_values("海波高度(m)", ascending=False)      # 降序排序
# 将 df_eares 首行的浮标 ID 存入变量 fbid
dffb = ② D df_anomaly中筛选出fbid浮标的所有数据
df1 = dffb.groupby("月", as_index=False)["海波高度(m)"].count()      # 分组统计次数
df2 = df1.sort_values("海波高度(m)", ascending=False)
# 将 df2 首行的月份存入变量 m, 代码略
print("海波高度异常记录最多的月份是:", m)
程序中①②处可选的代码有:
```

- A. df_anomaly.groupby("浮标 ID", as_index=False).count()
- B. df_eare[df_eare["浮标 ID"]==fbid]
- C. df_anomaly.groupby("经度", as_index=False).count()
- D. df_anomaly[df_anomaly["浮标 ID"]==fbid]

- (2) 为分析重点关注区域，取出某天海波高度最高的前 k 条数据存入列表 data 中，每个元素包含 2 个数据项，依次为浮标 ID（从 0 到 n-1）和海波高度。统计各浮标出现次数，将出现次数最多的浮标（若有并列，则取 ID 较小的浮标）所对应的区域列为重点关注区域，实现上述功能的部分 Python 程序如下：

```
tj = [0] * n      # n 为浮标个数
for d in data:
    tj[d[0]] += 1 tj 储存每个浮标出现的次数，data 中每个元素都包含浮标 ID 和海波高度2个数据项，d 为 data 中的元素，
    #统计每个浮标出现的次数
#找出次数最多的浮标 ID
maxfbh = 0
for i in range(1, len(tj)):
    
#输出浮标 ID 为 maxfbh 对应的区域，代码略
```

maxfbh 记录当前找到的出现次数最多的浮标 D, 对应的出现次数为 tj[maxfbh], 当 i 浮标出现次数更多, 则 maxfbh 更新为 i,

① 请在划线处填入合适的代码。

② 方框处的代码应为 C ▲ (单选: 填字母)

A. if tj[i] > maxfbh: maxfbh = tj[i]	B. if tj[i] > tj[maxfbh]: maxfbh = tj[i]	C. if tj[i] > tj[maxfbh]: maxfbh = i
---	---	---

15. 某公司使用一辆智能无人车来完成多件货物的运载任务(单件货物不拆分运输)。无人车每天单向行驶一次, 自起点依次经过装卸点 1, 装卸点 2, ..., 装卸点 n。每件货物的运载任务包括货物编号、货物重量(整数)、装货点和卸货点, 如第 15 题图所示:

货物编号	货物重量	装货点	卸货点
0	2	1	15
1	6	3	4
2	3	1	9
3	7	7	9
4	4	9	13

第 15 题图

规定: 每件货物的卸货点编号 > 装货点编号。请回答下列问题:

12

(1) 若一辆空车要当天完成如第 15 题图所示的运载任务, 则该车的最小载重量为 ▲ (填数字)。

(2) 若公司要在 D 天内将多件货物从同一装货点运送到各个目标卸货点。要求无人车按货物编号顺序依次完成所有货物运输, 求该车的最小载重量。实现上述功能的 Python 程序如下:

```
def canShip(weights, D, C):  
    days = 1  
    cur = 0  
    for w in weights:  
        if cur + w <= C:  
            cur += w  
        else:  
            days += 1  
            cur = w  
        if days > D:  
            return False  
    return True
```

要确定无人车当天完成运输的最小载重量, 需分析同一时段车上的货物总重量最大值。

货物0(装1卸15)、2(装1卸9)、3(装7卸9)、4(装9卸13)的装卸载过程中, 同时在车的货物重量峰值为:

货物0(重量2)+货物2(重量3)+货物3(重量7)=12

当到达装卸点7时, 货物0、2已装载, 货物3开始装载, 此时货物总重量最大。

非空列表 weights 按编号依次存放各件货物的重量, D 存放天数, 代码略

ans = max(weights) # ans 存放列表 weights 的最大值

while not canShip(weights, D, ans):

 ans = ans + 1

print(ans)

输出最小载重

① 若 `weights` 为 `[5,6,6,7,8,4,3]`，`D` 为 5，则该车的最小载重为 11 ▲ （填数字）。

② 将函数 `canShip` 改写如下，要实现相同功能，请在补充划线处代码。

```
def canShip(weights, D, C):
```

```
    days = 1
```

```
    cur = C
```

```
    for w in weights:
```

```
        if cur >= w:
```

```
            cur -= w
```

```
        else:
```

```
            days += 1
```

```
            if days > D:
```

```
                return False  
                cur=C-w ▲
```

```
    return True
```

`weights=[5,6,6,7,8,4,3]`，`D=5`，需按顺序分配5天运输，求最小载重量

尝试载重量为8:第1天5，第2天6，第3天6，第4天7，第5天8，第6天4+3=7→不可行。

尝试载重量为9:第1天5，第2天6，第3天6，第4天7，第5天8，第6天4+3=7→不可行。

尝试载重量为10:第1天5，第2天6，第3天6，第4天7，第5天8，第6天4+3=7→不可行。

尝试载重量为 11:第1天5+6=11，第2天 6，第3天7，第4天8+3=11，第5天4→共5天，符合要求。

第一个自定义函数的处理逻辑为当前载重 `cur` 剩余量足够则装货，否则新启一天。改写后 `cur` 初始为`C`(载重量)，不足时需要将`cur`重置为`C-w`，即新一天的剩余量。

(3) 公司要一天内完成多件货物的运输任务，任务列表 `tasks` 的每个元素为 `[sno, num, start, end]`，表示某件货物编号 `sno`、重量 `num`、装货点编号 `start` 和卸货点编号 `end`。求无人车的最小载重量。请在划线处填入合适的代码。

```
def insertInSorted(a, event):
```

```
    i, j = 0, len(a)-1  
    x=event[0] ①
```

函数需对 `event`(格式为[位置,重量变化])按位置排序，因此先初始化`x=event[0]`(按位置排序的关键字)

```
    while i <= j:
```

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

```
        if a[m][0] < x:
```

```
            i = m + 1
```

```
        elif a[m][0] == x and a[m][1] <= event[1]:
```

```
            i = m + 1
```

```
        else:
```

```
            j = m - 1
```

#在列表 `a` 中下标为 `i` 的位置插入元素 `event`；若 `i ≥ len(a)`，则在 `a` 末尾添加元素 `event`

```
    a.insert(i, event)
```

```
    return a
```

```
def assign(tasks):
```

```
    a = []
```

```
    for t in tasks:
```

```
        a = insertInSorted(a, [t[2], t[1]])
```

```
        a=insertInSorted(a, [t[3], -t[1]]) ②
```

```
    return a
```

`assign` 函数需将每个货物的装货事件[位置 `star`,重量]和卸货事件[位置 `end`, -重量]插入列表 `a`，因此在插入装货事件后，还需插入卸货事件

```
def proc(a):
    cur = 0
    maxw = 0
    for data in a:
        cur+=data[1]
        ③
    if cur > maxw:
        maxw = cur
    return maxw

# tasks 列表存放多件货物运输任务
a = assign(tasks)
capacity = proc(a)
print("无人车的最小载重量为: ", capacity)
```

proc函数需遍历事件列表，计算当前车上的总重量cur，并记录最大值maxw。每个 data 是[位置,重量变化]，因此需更新 cur 为 cur+data[1]