

高三下信息选考练习卷2

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 4 题

大模型（Large Model）是指具有大规模参数和复杂计算结构的机器学习模型。例如 OpenAI 的 GPT-4 模型。大模型的应用有自然语言处理、计算机视觉、语音识别和推荐系统等。大模型通过训练海量文本、图像、音频等数据来学习复杂的模式和特征，具有强大的泛化能力，可以对未见过的数据做出准确预测。

1. 下列关于信息和数据的描述，正确的是 **D**
A. 脱离语境的数据，也包含了各种信息 **脱离语境的数据是单纯的数据是没有意义的**
B. 同一组数据对不同的模型而言，所含价值和意义相同
C. 训练大模型的数据和大模型能处理的数据都是结构化数据
D. 数据所包含的信息能被大模型处理和学习，体现了信息的可加工处理性
2. 对于大模型和其训练所用的大数据，下列说法正确的是 **B**
A. 大模型训练用的大数据都需要人工采集并输入给大模型
B. 用大模型生成网络趣图等应用时，也要遵守信息社会道德准则和法律法规
C. 用大模型做商品智能推荐时，无需分析用户浏览、购买等行为数据的关联性 **X**
D. 大模型中的数据不受信息泄露、操作失误、存储器损坏等安全因素影响
3. 根据材料描述，下列选项中有可能未使用大模型人工智能应用的是 **C**
A. 输入文章概括其中心思想
B. 输入人像照片确定人物年龄
C. 输入关键词显示相关商品 **数据库检索**
D. 输入歌曲音频确定其歌名
4. 对于大模型中的数据编码，正确的是 **B**
A. 大模型中的数据以二进制或十六进制形式存储和处理的
B. 若其数据格式都是 ASCII 字符，则 1MB 数据有 2^{20} 个字节 **ASCII 占 1 个字节，1MB = 2^{20} B**
C. 若用 4 个字节来表示一个整数，则最大整数可以是 2^{32} **$2^{32} - 1$**
D. 若用 44.1kHz 采样频率，16bit 量化其音频数据，则 1 秒单声道音频约占 0.67MB

阅读下列材料，回答第 5 至 6 题

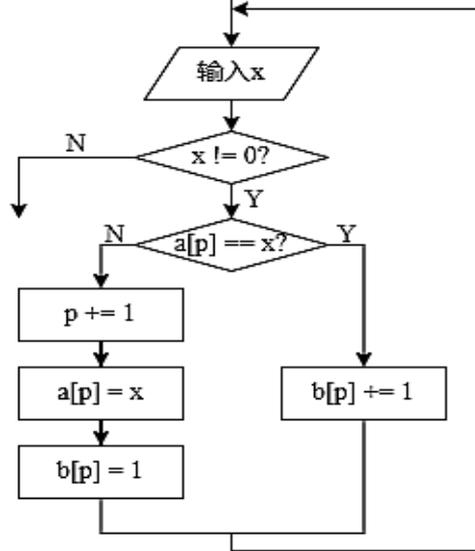
$$\frac{44.1 \times 10^3 \times 16 / 8 \times 1 \times 1}{1024 \times 1024} = 0.084M$$

某智能儿童手表采用鸿蒙操作系统设计。在数量众多的 APP 的支持下，连入厂家服务端，手表具有定位、通话、社交互动、健康监测、学习和娱乐等丰富功能。尤其是其离线定位功能，在附近其他相关设备组成的“查找”网络功能支持下，可以在信号盲区实现位置查找，确保孩子能被定位。

5. 关于该儿童手表的软件和硬件，下列说法不正确的是 **D**
A. 鸿蒙操作系统属于系统软件
B. 健康监测等智能性需要传感器支持
C. 儿童手表可以视为信息系统的一种终端
D. 有了网络存储，可以去掉手表内的存储器
6. 关于该儿童手表相关的网络或网络设备，下列说法正确的是 **A**
A. 该儿童手表需要能连接移动通信网或者连接 Wi-Fi 等联网功能
B. 在离线定位功能支持下，该手表无需连接任何设备就能定位 **X**
C. 在与网络同伴手表之间社交互动时，数据传输必须支持 HTTP 网络协议 **TCP/IP**
D. 在手表上开启 APP 连入服务端进行学习，该学习系统必然是基于 B/S 架构的

7. 某算法部分流程图如第7题图所示, 若 a、b 所有元素初始值都是 0, p 的初始值也是 0, 依此输入 580、570、565、564、564、520、520、520、0 后, a[4] 和 b[4] 的值分别为 **B**

当 a[p] 等于 x 时, b[p] 记录这个位置上的数有多少个。



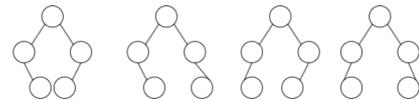
	0	1	2	3	4	5
a	0	580	570	565	564	520
b	0	1	1	1	2	3

第7题图

- A. 520 3 B. 564 2 C. 520 2 D. 0 1

8. 包含 5 个节点的二叉树, 其根节点的左右子树高度相同, 且中序遍历结果是“甲乙丙丁戊”, 那么其前序遍历结果不可能是 **D**

- A. 丙乙甲戊丁 左子树“甲乙”或“乙甲” B. 丙乙甲丁戊
C. 丙甲乙戊丁 右子树“丁戊”或“戊丁” D. 丙乙丁甲戊



9. 某浏览器的“后退”功能按钮用类似栈的数据结构实现 (假设栈容量为 3): 每打开新网站就会将上次的浏览记录压入“栈”中; 栈满时, 再浏览记录, 则从“栈”底删除较早的记录; 点击“后退”按钮时则从“栈”顶中弹出最近浏览的记录。若用一位数字表示每次打开的网站, 用大写字母“B”表示点击“后退”按钮, 如“1、2、B、3”操作后“栈”顶的网页是“1”。那么经过“1、2、3、4、B、5、6、B、B、B”操作后, 当前浏览的网页是 **A**

- A. 2 B. 3 C. 5 D. 1

10. 有如下 Python 递归程序:

```
def f(left, right):
    if left >= right:
        return left
    m = (left + right) // 2
    if key <= a[m]:
        return f(left, m)
    else:
        return f(m+1, right)
```

1、2、3、4 操作后, 当前浏览网页是4, 栈顶元素是3, 因此 B 操作会把网页3弹出并浏览。操作5、6过后, 当前浏览网页是6, 栈顶元素是5 (此时栈底的1会删除, 栈内是5、4、2)。后面的三次B操作依次弹出浏览的是5、4、2

相等时往前找

若 key=11, 下列选项中的数组 a 调用 f(0, 8) 的结果与其他三项不同的是 **C**

- A. a=[3, 3, 7, 7, 7, 9, 11, 11]
B. a=[3, 3, 9, 9, 9, 9, 16, 17]
C. a=[3, 3, 9, 9, 9, 11, 11, 17]
D. a=[3, 4, 6, 8, 8, 8, 11, 13]

11. 有如下 Python 程序:

```
res = ''; k = 1
for i in range(len(a)-2):
    if ord(a[i]) * k < ord(a[i+2]) * k:
        a[i], a[i+2] = a[i+2], a[i]
    k = -k
for s in a:
    res += s
print(res)
```

没有外循环, 所以只排一趟

奇偶分离排序
奇升, 偶降

已知数组 a 共 8 个元素, 每个元素都为单个字符, 运行该程序后输出结果可能是 **D**

- A. computer
- B. repomcut
- C. nesestic
- D. sciences

12. 已知链表 a 存放了学生 ID 和借阅图书书名, 链表 b 存放了学生 ID 和姓名。两链表节点都按学生 ID 升序存放, 如第 12 题图 (a) 和 (b) 所示。链表 a 的头指针指向额外的空元素。现要将两链表中 有共同学生 ID 的节点连接成一个新链表, 如第 12 题图 (c) 所示。若无共同 ID, 跳过

```
ha = q = 0
p = ha
while a[p][2] != -1:
    tp = a[p][2]
    if q == -1: b走完了
        a[p][2] = -1
        break
    if a[tp][0] == b[q][0]:
        a[tp].append(b[q][1])
        p = a[p][2]
    elif a[tp][0] > b[q][0]:
        q = b[q][2]
    else:
        # 共同ID内容, 此处需要删除没有共同ID的a链中的点
        a[p][2] = b[q][2]
    # 从 ha 的下一个位置开始输出链表 a,
    # 输出结果即为两表连接结果, 代码略
```

p->tp

b走完了

p=a[p][2]

a链ID值大于b链, b走

a链ID小于b链, a走, 为了让a链最后存放共同ID内容, 此处需要删除没有共同ID的a链中的点

找到共同ID的学生, B链姓名加入到a中, a链往后走一个节点

第 12 题图(a)

	ID	书名	指针
ha→0	-	-	1
1	6	三体	2
2	7	大秦	3
3	8	荆棘鸟	4
4	8	三体	5
5	10	红楼梦	-1

第 12 题图(b)

	ID	姓名	指针
q→0	6	张昀轩	1
1	8	王泽	2
2	9	林雯	-1

第 12 题图(c)

	ID	书名	指针	姓名
ha→0	-	-	1	-
1	6	三体	3	张昀轩
2				
3	8	荆棘鸟	4	王泽
4	8	三体	-1	王泽
5				

- A. ① q = b[q][2] ② p = a[p][2]
- B. ① p = a[p][2] ② a[p][2] = a[tp][2]
- C. ① tp = a[tp][2] ② p = a[tp][2]
- D. ① p = tp ② p = a[tp][2]

tp=a[tp][2]让 tp 指针移动, 而 p 指针不动, 那么下一次循环时又会执行“tp=a[p][2]”导致 tp 指针不变。因此这里适合让 p 指针移动, 进入下一次循环时, tp随着 p的移动而移动

二、非选择题（本大共 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. 某路口的交通信号灯控制系统每隔 1 秒采集路口车辆情况，若在 1 分钟以内（采集 60 次）检测到持续有车辆通过路口，且持续数量最大值超过 30 辆时视为大流量，需要设置绿灯持续 50 秒，黄灯持续 5 秒，红灯持续 45 秒。其他流量情况下设置绿灯持续 30 秒，黄灯持续 5 秒，红灯持续 25 秒。如 [2, 4, 0, 1, 0, 3] 表示检测了 6 次，每次检测到的车辆数分别是 2、4、0、1、0、3 辆车，最大持续通过的车辆数是 6 辆。请回答下列问题。

- (1) 若十次检测的车辆数据是 [3, 0, 2, 4, 0, 3, 1, 1, 0, 7]，则持续数量最大是 7。
(2) 实现上述功能的部分 Python 程序如下，请在划线处填入合适代码。

```
while True:
    cur = 0
    maxn = 0
    for i in range(60):
        # 接受一次系统中监控设备检测到的车辆数量，存入 n，代码略
        if n > 0:
            cur += n
            if cur > maxn: 或 cur >= maxn
                maxn = cur
        else:
            cur = 0 或 cur = n
            # 延时到 1 秒，代码略
            if maxn > 30:
                # 发送“大流量”状态信息，设置下一轮的绿灯时间 50 秒，
                # 黄灯时间 5 秒，红灯时间 45 秒，代码略
            else:
                # 发送“正常流量”状态信息，设置下一轮的绿灯时间 30 秒，
                # 黄灯时间 5 秒，红灯时间 25 秒，代码略
            # 等待当前一轮红绿灯结束，进行下一轮红绿灯设置，代码略
```

14. 为了提升图书馆、博物馆等场馆服务质量，部分公共场馆采用了智能预约与核验系统。用户通过网络平台或者小程序，使用个人身份信息进行提前或现场预约场馆。用户入闸核验时，智能终端识别用户身份证或者预约二维码，并通过无线通信方式将数据传输到 Web 服务器，服务器在数据库中查找用户身份信息后做出是否打开闸机放行的处理。请回答下列问题。

- (1) 相比较而言，以下身份认证方式中，用户通行效率较低的是 A（单选，填字母：A. 用户名+口令 / B. 身份证刷卡 / C. 预约二维码扫码）
(2) 通过手机 APP 预约成功后，用户持身份证入闸核验时发现道闸并未打开放行，下面描述中，不可能引起该故障的是 C（单选，填字母）
A. 身份证识别传感器故障
B. 道闸机械故障等原因，无法完成执行器的控制
C. 用户手机等电子设备未接入场馆网络，造成无法访问

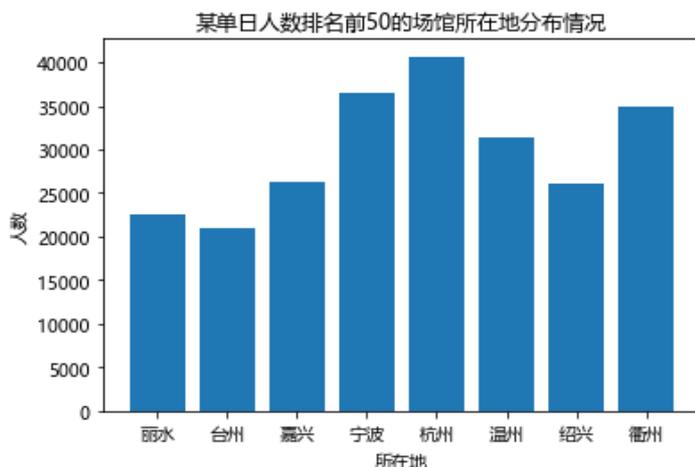
- ① 对关键数据，如个人敏感数据加密存储与传输；② 为服务器安装病毒防治软件，降低病毒感染风险；③ 安装并开启防火墙，防止黑客和病毒利用漏洞入侵系统；④ 对不同用户分配不同权限，利用访问控制技术限制数据资源；⑤ 建立磁盘阵列，及时做好数据备份

(3) 下列关于该系统的设计与实现说法，正确的有 BC (多选，填字母)。(注：全部选对的得 2 分，选对但不全的得 1 分，不选或有错的得 0 分)

- A. 预约二维码的识别需要 RFID 识别技术的支持
 B. 用户在网上预约场馆成功后，身份信息要保存在服务器数据库中
 C. 可以使用 Flask Web 框架和 SQLite 3 编写服务器程序
 D. 系统完成后需要进行软硬件测试，而文档编写可以忽略
- (4) 为了增加信息系统安全性，请从保护系统数据不丢失或保护用户个人敏感信息不泄漏角度，请列举两种方法。
- (5) 小林整理出了某些时间的场馆访问人数等数据，如第 14 题图 a 所示。现要统计“2024-08-03”单日访问人数最多的前 50 个场馆的所在地归属，并绘制如第 14 题图 b 所示的柱形图。部分实现程序如下所示。

	A	B	C	D
1	名称	所在地	时间	人数
2	上城区文化馆	杭州	2024-08-03	1720
3	东阳市博物馆	金华	2024-08-03	866
4	东阳市图书馆	金华	2024-08-03	1025
5	中国数安港城市书房	温州	2024-08-03	1501
6	中国丝绸博物馆	杭州	2024-08-03	2014
7	中国木雕博物馆	金华	2024-08-03	874
8	南门街自助借阅区	金华	2024-08-03	56
4579	龙川路悦读吧	金华	2024-08-25	929
4580	龙泉市博物馆	丽水	2024-08-25	1724
4581	龙泉市文化馆	丽水	2024-08-25	1869
4582	龙游县博物馆	衢州	2024-08-25	1915

第 14 题图 a



第 14 题图 b

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
df = pd.read_excel("部分场馆访问记录.xlsx")
```

```
df1 = df[df["时间"]=="2024-08-03"]
```

```
plt.bar(df1["所在地"], df1["人数"])
```

排序，前50，归属地

单日访问人数最多的前50个场馆的所在地归属

方框中应填入的语句依次为_____（选3项，填数字序列，少选、多选、错选或次序错均不得分）。

- ① `df1 = df1.head(50)` **4、1、3**
② `df1 = df.groupby("所在地", as_index=False).sum()`
③ `df1 = df1.groupby("所在地", as_index=False).sum()`
④ `df1 = df1.sort_values("人数", ascending=False)`

(6) 为了研究哪些场馆人流量较少，需要在第14题图a所示的数据集中筛选出单日人数小于100的数据以便进一步研究分析，可在第(5)小题的基础上再执行如下语句，请在划线处填入合适的代码。

```
df2 = df[df["人数"]<100]    或    df[df.人数<100]
```

15. 某机场共有 m 个登机口，通过 m 个廊桥连接到飞机舱门，登机口编号和廊桥编号相同，范围都是 $0 \sim m - 1$ 。现有 n 架航班要停靠机场接受登机，廊桥分配程序要为每架飞机分配一个合适的廊桥，并在对应的登机口显示当前停靠的航班信息。已知飞机的到达时刻与占用廊桥的时长（示例如第15题图a所示）。飞机到达时刻是按时间先后顺序给出的，且同一时刻最多有一架飞机到达。前一班飞机离开廊桥与后一班接管廊桥之间的时间间隙忽略不计。对于一架到达的飞机，程序会优先分配目前未占用的廊桥；若所有廊桥都被占用，则优先分配等待时间最短的；若有多个可分配的廊桥（如：多个未被占用或等待时间相同），程序会任意分配一个廊桥。

飞机编号	到达时刻	占用时长	航班号
0	1	4	MF8556
1	2	9	KN5956
2	4	5	MU5186
3	5	3	HU7610
4	7	5	CA8685

第15题图a

```
0号登机口: MF8556, HU7610, CA8685
1号登机口: KN5956
2号登机口: MU5186
```

当登机口数量 $m = 3$ 时，各廊桥可能的分配情况。注意其中已离开廊桥的航班信息未清除。

第15题图b

请回答以下问题。

(1) 如第 15 题图 a 所示, 若 0 号航班“MF8556”需要占用廊桥的时长是 6, 保持 $m = 3$ 和其他条件不变, 且前三架飞机的分配方案也不变, 那么 4 号航班“CA8685”的登机口是 2 号 (选填数字: 0/1/2)。

(2) 已知航班占用廊桥的情况按航班占用廊桥的结束时间从小到大的顺序保存在了队列 work 中, 每个元素包含两个数据区域: 廊桥占用的结束时刻和廊桥编号。如第 15 题图 a 所示, 前三架飞机到达后 $work = [[5, 0], [9, 2], [11, 1]]$ 。push 函数将新元素插入到队列的合适位置, 请将划线处程序补充完整。

```
# endtime 和 gid 是待插入元素的结束时刻与廊桥编号
# head 指向队首元素、tail 指向队尾的下一个位置
def push(endtime, gid, head, tail):
    j = tail - 1
    work.append([])
    tail += 1
    while j >= head and work[j][0] > endtime:
        work[j+1] = work[j]
        j -= 1
    work[j+1] = [endtime, gid]
    return tail
```

插入排序,
关注: 边界问题,
比较对象问题

(3) 主程序。假设程序要为 n 架停靠机场的飞机分配廊桥 (飞机编号为 $0 \sim n - 1$), 以下程序实现了廊桥分配功能, 请将划线处程序补充完整。

```
gate = [] # gate[i] 保存第 i 号登机口停靠过的航班信息
work = [] # 保存被占用的廊桥信息
free = [] # 保存空闲廊桥的编号
headw = tailw = 0
headf = tailf = 0
for i in range(m):
    gate.append([])
    free.append(i)
    tailf += 1
for i in range(n):
    # 输入第 i 架飞机的到达时刻, 占用廊桥时长, 代码略
    while headw != tailw and work[headw][0] <= s:
        free.append(work[headw][1])
        tailf += 1
        headw += 1
    if headf != tailf:
        gid = free[headf]
        headf += 1
        gate[gid].append(i)
        tailw = push(s + t, gid, headw, tailw)
    else:
```

或 $headw < tailw$ and $work[headw][0] <= s$
整理 i 架飞机到达时, 空闲廊桥的情况, 前面飞机占用的廊桥如果此时闲置, 就加入到 free 队列, 注意: 队列检测必须有的条件“队不为空”

如果有空闲廊桥
出队

如果没有空闲廊桥，就在占用廊桥里找最早结束的

```
cur = work[headw] 最早结束的廊桥的信息：[结束时间, 编号]
```

```
headw += 1
```

```
gate[cur[1]].append(i)
```

```
tailw = ③push(cur[0] + t, cur[1], headw, tailw)
```

第 i 架飞机分配好廊桥后在对应停靠的登机口更新显示航班信息，代码略

入队元素的信息[结束时间，编号，工作队列队首，工作队列队尾]，插入到原工作队列排序