

2025年6月真题解析

1. 某校师生可在学校图书馆自助设备上借还图书，还可通过在线图书馆系统查看借阅记录、续借图书、浏览新书推荐。下列关于数据的说法，正确的是 (C)

- A. 图书的封面图像和馆藏数量在编码方式上没有差异
- B. 数据库的应用降低了图书数据管理的效率
- C. 对借阅数据的加工处理可为图书采购提供依据
- D. 图书借还数据仅存储在自助设备中

图书封面图像(通常采用基于像素的图形编码方式)与馆藏数量(采用数值编码方式)在数据编码原理上存在本质差异。

2. 某校师生可在学校图书馆自助设备上借还图书，还可通过在线图书馆系统查看借阅记录、续借图书、浏览新书推荐。下列措施中，不能有效提升在线图书馆数据安全的是 (A)

- A. 向用户发送借阅到期的提醒信息
- B. 对用户信息进行加密存储
- C. 定期修改管理员密码
- D. 为系统服务器增加不间断电源

不间断电源(UPS)在断电时提供临时电力，防止服务器突然关机导致数据丢失或损坏。确保数据的可用性和完整性，是物理层安全的重要措施。

3. 某智慧公交系统中的车载终端内置了北斗定位、4G/5G通信、音频采集、NFC识别等模块，实时采集、处理公交车辆行驶数据，然后传输至服务器；车载摄像头识别违规驾驶行为，发出语音提醒，并通过车载终端将违规视频传输至服务器；公交APP为用户提供查询服务，还可在电子地图上实时显示公交车辆行驶路线和位置。在电子地图上实时显示公交车辆行驶路线和位置的过程中，没有用到的技术是（D）

A. 北斗定位

B. 4G/5G通信

C. 数据可视化

D. NFC

4. 某智慧公交系统中车载终端内置了北斗定位、4G/5G通信、音频采集、NFC识别等模块，实时采集、处理公交车辆行驶数据，然后传输至服务器；车载摄像头识别违规驾驶行为，发出语音提醒，并通过车载终端将违规视频传输至服务器；公交APP为用户提供查询服务，还可在电子地图上实时显示公交车辆行驶路线和位置。下列关于公交APP功能和设计的说法，正确的是（A）

A. 该APP具有数据处理功能

B. 该APP的运行不需要操作系统支持

C. 该APP升级和维护都是为了适应公交线路的变化

D. 该APP只能直接从车载终端获取车辆实时位置

B. 错误，APP必须依赖操作系统(如Android/iOS)运行。

C. 错误，升级和维护可能涉及功能优化、性能提升、安全修复等，不限于线路变化。

D. 错误，APP从服务器获取数据，而非直接从车载终端获取。

5. 某智慧公交系统中的车载终端内置了北斗定位、4G/5G通信、音频采集、NFC识别等模块，实时采集、处理公交车辆行驶数据，然后传输至服务器；车载摄像头识别违规驾驶行为，发出语音提醒，并通过车载终端将违规视频传输至服务器；公交APP为用户提供查询服务，还可在电子地图上实时显示公交车辆行驶路线和位置。下列关于该系统中硬件和网络的说法，

正确的是（ B ）

A. 该系统无需在公交车上配备输出设备

B. 车载终端中必定有处理器部件

C. 车辆行驶数据传输至服务器无需网络协议的支持

D. 通过4G/5G网络才能使用公交APP的查询功能

A. 该系统识别违规驾驶行为发出语音提醒，因此配备了输出设备；

C. 数据传输到服务器需要网络协议；

D. 除了4G/5G也可以借助计算机网络使用APP。

6. 某智慧公交系统中的车载终端内置了北斗定位、4G/5G通信、音频采集、NFC识别等模块，实时采集、处理公交车辆行驶数据，然后传输至服务器；车载摄像头识别违规驾驶行为，发出语音提醒，并通过车载终端将违规视频传输至服务器；公交APP为用户提供查询服务，还可在电子地图上实时显示公交车辆行驶路线和位置。将车载终端采集的声音存储为未经压缩的Wave格式音频文件，下列说法不正确的是（C）

A. 声音采集实现了从模拟信号到数字信号的转换

B. 音频采集模块的采样频率会影响音频的音质

C. 现场声音越嘈杂，得到的音频文件存储容量越大

D. 为了节省存储空间，可将Wave格式音频转换为MP3格式

A.正确，音频采集模块将模拟声音信号转换为数字信号。

B.正确，采样频率越高，音质越好，但文件越大。

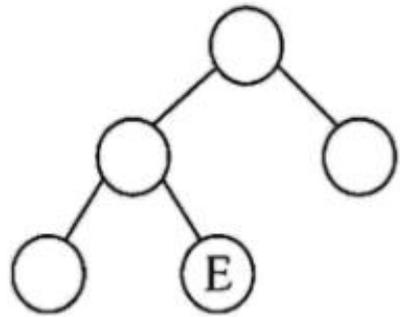
C.错误，存储容量由采样率、位深、声道数和时间决定，与声音嘈杂程度无关。

D.正确，MP3 是有损压缩格式，可显著减小文件大小。

7. 某智慧公交系统中的车载终端内置了北斗定位、4G/5G通信、音频采集、NFC识别等模块，实时采集、处理公交车辆行驶数据，然后传输至服务器；车载摄像头识别违规驾驶行为，发出语音提醒，并通过车载终端将违规视频传输至服务器；公交APP为用户提供查询服务，还可在电子地图上实时显示公交车辆行驶路线和位置。车载摄像头识别违规驾驶行为是基于神经网络方法实现的，下列说法不正确的是（C）

- A. 识别违规驾驶行为是人工智能技术的应用 A.正确，基于神经网络的识别属于人工智能。
- B. 训练神经网络模型时需要提供驾驶行为数据 B.正确，模型需通过大量标注数据训练。
- C. 进行违规驾驶行为识别时仍离不开原始训练数据 C错误，识别时依赖训练好的模型，而非原始训练数据。
- D. 识别违规驾驶行为的结果并不总是正确的 D.正确，模型可能因数据偏差或复杂场景出现误判。

8. 某二叉树如图所示，E节点在前序遍历序列中的位置记号为x。下列二叉树中，E节点在中序遍历序列中的位置序号也为x的是（ A ）



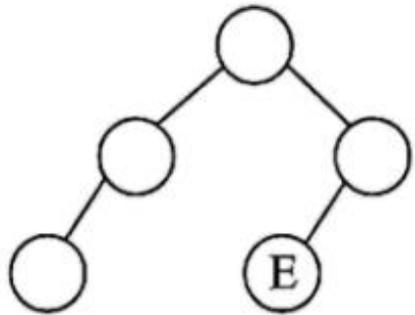
000E0 X=4

000E0

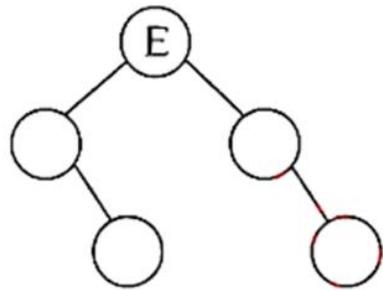
00E00

000E0

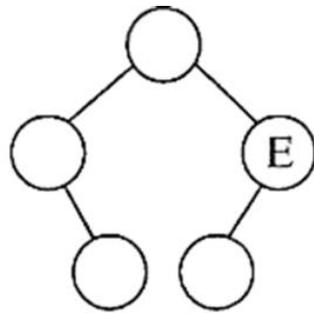
00E00



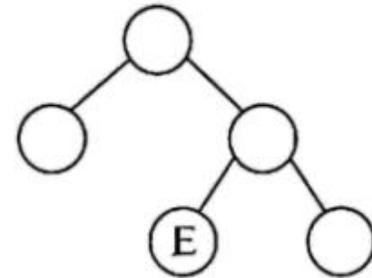
A.



B.



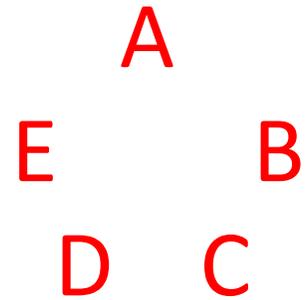
C.



D.

9. 某队列中，队首到队尾的元素依次为A, B, C, D, E。元素出队后直接输出或重新入队，若输出次序为B, D, C, E, A, 则元素A重新入队的最少次数为 (B) **画圈法**

- A. 1 B. 2 C. 3 D. 4



10. 有如下Python程序段:

```
i,r = n , ""
```

```
while i < len (s) :
```

```
    r += s[i]
```

```
    i += 1
```

```
    if i % 5 == 0:
```

```
        i += n
```

循环变量 i 的初值为2，因此变量r的首个字符肯定是“c”。
可以排除选项 AB。当 i=4 时，即取得字符“e”后,i 增加1
后变为5，此时i%5==0 成立，执行代码 i+=n,i=7,因此字
符“fg”都被跳过，选 C。

若s为“abcdefghi”， n为2，运行该程序段后， r的值为 (C)

- A. "abefi" B. "abfg" C. "cdehi" D. "cdhi"

11. 有如下Python程序段:

```
tag=[0]*len(data)
```

```
p=i=0
```

```
while i<len (data) :
```

```
    if tag[p]==0 and data[p][1]!=-1:
```

```
        tag[i]+=1
```

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

```
    else:
```

```
        tag[i]+=tag[p]
```

```
    i+=1
```

```
    p=i
```

若data为[[11,3],[23,-1],[15,0],[26,1],[63,2]], 运行该程序段后, tag[4]值为 (D)

A. 1 B. 2 C. 3 D. 4

该题的程序功能是纪录由链表中每个节点到链表末尾的步数。

假设以i节点为链头, 遍历链表求得步数(长度), 步数存储在tag[i]中。

如果在遍历过程中发现当前节点(p)已经访问过 (tag[p]!=0), 则不再需要遍历到链表结尾, 直接将tag[p]累加到tag[i]中。

程序功能: 在非环链的普通链表中快速找链头。

0 1 2 3 4

data为[[11,3], [23,-1], [15,0], [26,1], [63,2]]

	0	1	2	3	4
tag[i]					

用列表模拟的链表中，列表中所有元素组成链，且不存在环链。

如何快速找到链头？

```
data = [[11, 3], [23, -1], [15, 0], [26, 1], [63, 2]]
```

```
trace = {}
```

```
for i in range(len(data)):
```

```
    trace[data[i][1]] = i
```

```
t = -1
```

```
while t in trace:
```

```
    print(t)
```

```
    t = trace[t]
```

```
print("链表的head为", t)
```

如果存在多链，如何找所有的链头？

binary

12. 定义如下函数，返回key在列表data中的下标，若key不存在，则返回-1。

```

def bsearch(data,key):
    i,j=0,len(data)-1
    while i<=j:
        m=(i+j)//2
        for k in range(m-1,m+2):
            if i<=k<=j and data[k]==key:
                return k
        if data[m] < key:
            i=m+2
        else:
            j=m-2
    return -1

```

- A. [16,19,18,22,20,29,25]
- B. [16,19,20,18,22,25,29]
- C. [18,16,19,20,22,29,25]
- D. [18,16,20,19,25,22,29]

对分查找的变式，是以3个数一组为进行对分查找。组内乱序，组间相对有序。
 当列表长度较小时，不一定是3个一组。
 程序在对分时，每查找一个m位置时，会利用for循环遍历m-1、m、m+1三个位置，若不存在，则会根据key值向左右方向缩小查找范围，因此，数据必须具备以下特征，data[0]和data[1]数据必须比data[3]小，data[5]、data[6]数据必须比data[3]大，选项B错在数据19，若key=19时，是无法查到的，因此答案为B。

如果调用函数返回结果不正确，则data可能是 (B)

13. 某小组搭建水质监测系统，采集某水域溶解氧和pH的数据，进行水质监测。对于每个传感器，智能终端每小时获取3次数据，将3个数据的中位数（排序后处于中间位置的数）通过5G模块上传至服务器。服务器检测到异常情况时，向管理员发送警示信息，并通过智能终端控制指示灯闪烁。用户通过浏览器可查看系统数据。请回答下列问题：

(1) pH数据从采集到存入数据库的数据流向为 **B** （单选）。

- A. 传感器→服务器→智能终端→数据库
- B. 传感器→智能终端→服务器→数据库

(2) 该系统的数据处理 **C** （单选）

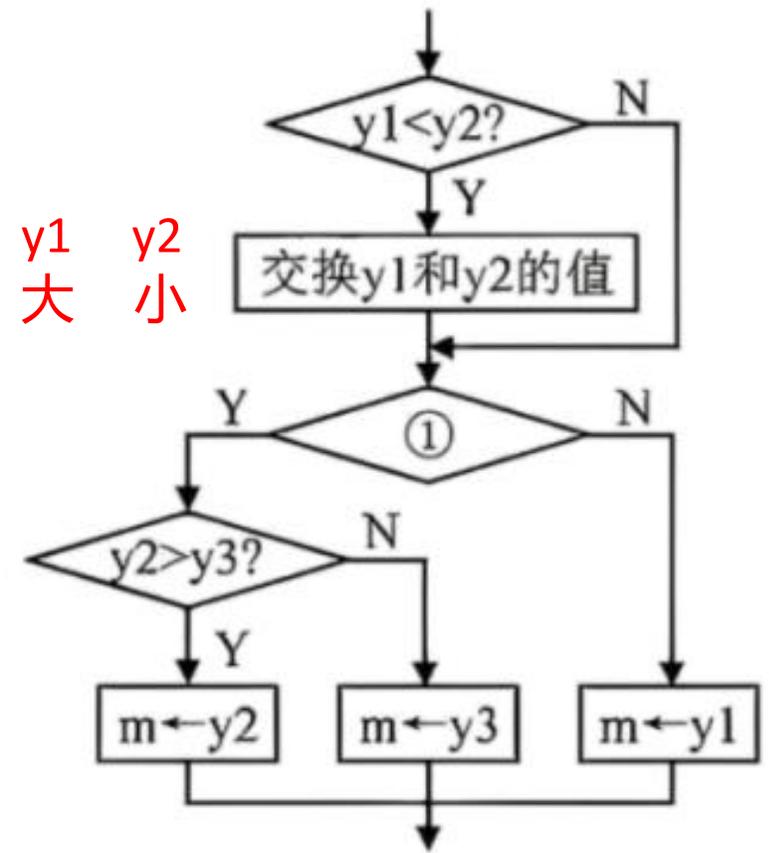
- A. 全部在服务器端完成
- B. 全部在智能终端完成
- C. 部分在智能终端完成，部分在服务器端完成

注意：此类题一定要看题干获取相关信息，题干中智能终端不需接收服务器的指令，只需要上传中位数

(3) 若连接在智能终端上的5G模块突发故障不能工作，会引发的问题有 **BC** (多选)

- A. 无法通过浏览器访问溶解氧历史数据
- B. 智能终端无法传输pH数据至服务器
- C. 服务器向智能终端传送控制信号失败
- D. 服务器向管理员发送警示信息失败

(4) 智能终端每小时获取的3个pH数据分别存入 y_1 、 y_2 和 y_3 ，将中位数存入 m 的部分流程图如图所示。图中①处应填入 **$y_1 > y_3?$** 。



y_1 大
 y_2 小

答题注意：缺少“？”不给分!!! 流程图题一定要观察图中其他语句的书写形式，做到格式统一

(5) 现需增加水温监测的功能，在智能终端接入温度传感器后，还需对软件部分作多处修改。请用文字描述其中1处修改建议。

传感器 → 智能终端(采集、计算中位数) → 5G网络 → 服务器(接收、存储、分析)

- ①在智能终端处修改代码，增加(烧录)获取/采集温度传感器数据的代码(程序)；
- ②在智能终端处修改代码，增加(烧录)向服务器上传温度传感器数据的代码(程序)；
- ③在服务器端处修改代码，增加接收温度传感器数据并上传至数据库的代码(程序)；
- ④在数据库中增加记录温度的字段并将温度数据写入/录入。

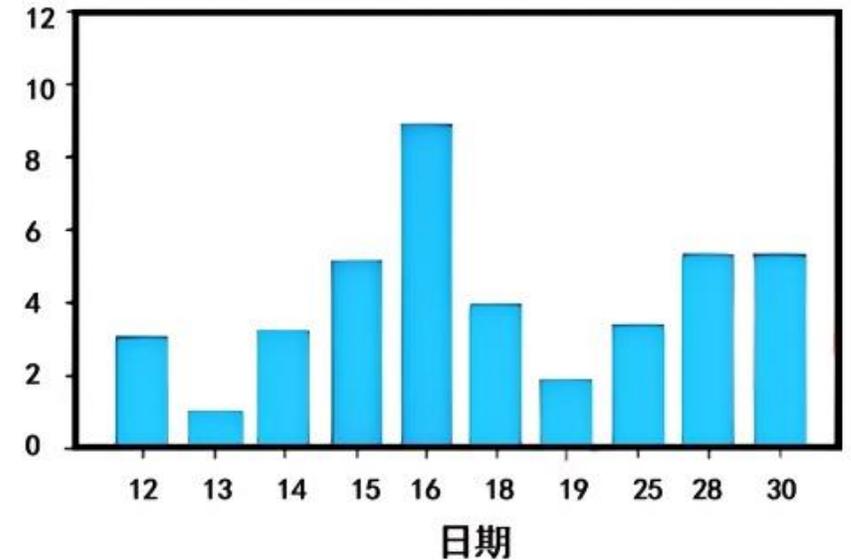
要点:明确对象(智能终端/服务器/数据库)+修改代码+简要描述(侧重数据流向而非数据处理，功能描述不给分)。说明:只需要写出一点修改建议。

14. 水质监测系统已采集了某水域一年的pH数据，该水域pH值的正常范围为6-8。现要对这些数据进行分析，请回答下列问题：

监测点	月	日	时	PH
监测点1	1	1	0	7.865
监测点1	1	1	1	7.724
监测点1	1	1	2	8.021
监测点1	12	31	21	7.129
监测点1	12	31	22	7.235
监测点1	12	31	23	7.214

图a

3月PH大于8的日分布情况



图b

(1) 将监测点1的数据导出，存于pHdata.xlsx文件中，如图a所示。现要找出pH均值最高的月份，并统计该月pH大于8的日分布情况，绘制如图b所示的柱形图。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("pHdata.xlsx")

df1=df.groupby("月",as_index=False).pH.mean() #分组求平均

df2= ① **E**

#将df2首行的月份存入m,代码略

print("pH值最高月份序列为:",m)

df_m= ② **A**

df_ex=df_m[df_m["pH">8] #筛选

df_cnt= ③ **D**

#重命名pH列名称为"次数",代码略

plt.bar(df_cnt["日"],df_cnt["次数"])

#设置绘图参数，显示如图b所示的柱形图，代码略

• C.df.sort_values("pH",ascending=False)

E.df1.sort_values("pH",ascending=False)

A.df[df["月"]==m]

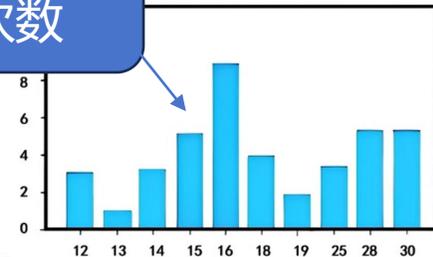
B.df2[df2["月"]==m]

D.df_ex.groupby("日",as_index=False).pH.count()

F.df_ex.groupby("时",as_index=False).pH.count()

1天中pH>8
的次数

pH大于8的日分布情况



监测点	月	日	时	PH
监测点1	1	1	0	7.865
监测点1	1	1	1	7.724
监测点1	1	1	2	8.021
监测点1	12	31	21	7.129
监测点1	12	31	22	7.235
监测点1	12	31	23	7.214

程序中①②③处可选代码有:

A.df[df["月"]==m]

B.df2[df2["月"]==m]

C.df.sort_values("pH",ascending=False)

#降序排序

D.df_ex.groupby("日",as_index=False).pH.count()

#分组计数

E.dfl.sort_values("pH",ascending=False)

F.df_ex.groupby("时",as_index=False).pH.count()

(2) 将7-12月的pH数据存储于列表data中，要求出一个**最长连续**序列，其中每个pH值均在正常范围内。如果这样的序列有多个，则选择**数值总和最小**的序列（若仍有多，选择**最早**出现的），输出其长度和起始下标。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pH数据，按采集的时间顺序存储于列表data中，代码略

```
maxn=start=maxt=0
```

```
① i=0
```

```
while i < len (data) :
```

```
    if 6<=data[i]<=8:
```

```
        total=0
```

```
        k=i k表示连续区域的开始
```

```
        while i < len (data) and 6 <= data[i] <= 8:
```

```
            total+=data[i] 连续区间，值累加
```

```
            i+=1
```

```
② n=i-k #跳出while时，说明连续段结束，计算连续的长度
```

```
    if n>maxn:
```

```
        maxn=n
```

```
        start=k
```

```
        maxt=total
```

```
    elif ③ n==maxn and total<maxt:
```

```
        start=k #长度相等时，选总和最小
```

```
        maxt=total
```

```
    i+=1
```

```
#输出最长连续序列的长度maxn和起始下标start，代码略
```

15. 某校实验室有多台相同的实验仪器，每台仪器同一时间仅限一位学生使用。现要开发一个仪器预约系统，开发前先对实验室的学生进出记录进行统计分析。请回答下列问题：

学号	1010	1021	3009	1010	2081	3009	1021
时间	08:00	08:05	08:20	08:45	08:50	08:50	09:15
行为	进	进	进	出	进	出	出

(1) 实验室某天09:30前学生进出情况如图a所示，09:00时实验室内的学生数为 2。

(3) 仪器预约系统中，学生可在线预约仪器使用时段，系统按预约提交顺序逐个处理，如果预约的使用时段有仪器可用，预约成功，否则失败。

函数与方法	功能
lst.insert(i,x)	在列表lst中下标为i的位置插入元素x。若i大于等于len (lst) ，则在lst末尾添加元素x。 lst[i:] 后移 lst:[10, 20, 30] ; lst.insert(1, 8) 结果: [10, 8, 20, 30]

函数参数data列表存放已按提交顺序排列的预约信息，每个元素包含4个数据项，依次为学号、起始时间、终止时间、是否成功。其中起始、终止时间都用8位数字字符串表示，如“09280830”表示9月28日8点30分，“是否成功”数据项用于存放每个预约的处理结果。

参数m存放实验室仪器台数。

"""

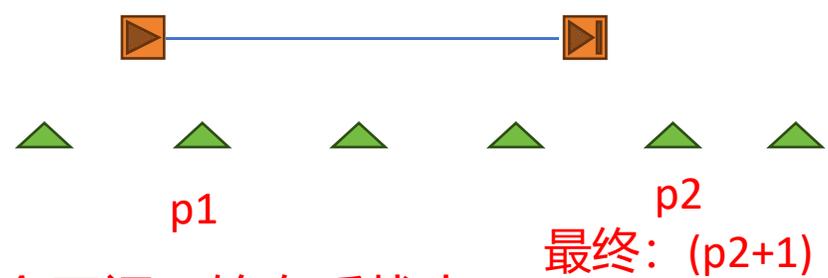
本题：1.a数据结构是什么？
 2.i=0的循环很关键！可以理解列表a结构
 3.p1与预约开始有关 p2与预约结束有关
 4.越来越多时间点加入a，但这些时间点看不出对应关系

```
def proc(data,m):
    a=[ ]
    for i in range(len(data)):
```

flag=True

```
    p1=0
    while p1 < len(a) and a[p1][0] <= data[i][1]:
        p1+=1
    if p1 > 0:
        p2=p1-1
    else:
        p2=0
```

停止时，p1走到列表末尾之后
 或a[p1][0]>data[i][1]
 目的是在a中找到第一个比
 data[i][1]时间点来的大时间点



p2是在a中p1的前一个区间开始向后找大于data[i][2]时间点

```
while p2 < len(a) and a[p2][0] < data[i][2]:
    if a[p2][1]==m 或a[p2][1]>=m:
        flag=False
        break
    p2+=1
    data[i][3]=False
```

p2在寻找过程中发现某时间仪器已满，就不找了 (break)，

if flag: 没有被停止，表示i记录可以预约，

```
for j in range(p1,p2):
    a[j][1]+=1 #批量设置 使用机器数+1
a.insert(p2,[data[i][2],0])
a.insert(p1,[data[i][1],1]) #推测a[j][1]是机器数量
if p1>0:
    a[p1][1]=a[p1-1]+1
a[p2+1][1]=a[p2][1]-1
```

```
data[i][3]=flag
```

常规解法：一条一条时间线，看重叠数量

data[i]=[6,0825,0855,False]

先在p2位置插入结束节点，然后在p1位置插入开始节点，此时p2位置的节点变成p2+1位置

在p1位置插入

在p2位置插入

