

高三信息技术专题：目标区间搜索

一、同值区间分析

已知一个列表 `data` 记录了一系列数值。请找出连续相等元素的最长子数组（区间），并计算该区间的长度和起始位置。

方法一：遍历计数法 <pre> data = [75, 76, 76, 75, 78, 80, 95, 95, 95, 82, 81, 79] ans = 1 c = 1 i = 1 while i < len(data): if data[i] == data[i-1]: c += 1 if c > ans: ans = c else: c = 1 i += 1 print("最长同值区间长度为： %d" % ans) </pre>	方法二：双指针法 <pre> data = [75, 76, 76, 75, 78, 80, 95, 95, 95, 82, 81, 79] ans = 1 ans_st = 0 st = 0 i = 1 while i < len(data): if data[i] == data[i-1]: current_len = i - st + 1 if current_len > ans: ans = current_len ans_st = st else: st = i i += 1 print("最长同值区间： 从第%d 个数开始，长度： %d 个数" % (ans_st+1, ans)) </pre>
--	--

二、范围区间分析

已知一个列表 `data` 记录了一系列数值，以及一个目标范围 `[low, high]`。请找出连续元素均落在该范围内的最长子数组（区间），并计算该区间的长度、起始位置，以及该区间长度占总长度的百分比。

方法一：内循环扩展法 <pre> data = [75, 76, 76, 75, 78, 80, 95, 95, 95, 82, 81, 79] low = 75 high = 85 ans = 1 ans_st = 0 i = 0 while i < len(data): if low <= data[i] <= high: st = i n = 1 while i + n < len(data) and low <= data[i+n] <= high: n += 1 if n > ans: ans = n ans_st = st i += n else: i += 1 print("最长目标区间： 从第%d 个数开始，长度： %d 个数" % (ans_st+1, ans)) print("区间占比： %.1f%%" % (ans / len(data) * 100)) </pre>	方法二：双指针滑动法 <pre> data = [75, 76, 76, 75, 78, 80, 95, 95, 95, 82, 81, 79] low = 75 high = 85 ans = 0 ans_st = 0 left = 0 for right in range(len(data)): if not (low <= data[right] <= high): left = right + 1 continue current_len = right - left + 1 if current_len > ans: ans = current_len ans_st = left print("最长目标区间： 从第%d 个数开始，长度： %d 个数" % (ans_st+1, ans)) print("区间占比： %.1f%%" % (ans / len(data) * 100)) </pre> 区间中断，左边界移到当前右边界的下一位
--	---

三、平滑区间分析

已知一个列表 data 记录了一系列数值，以及一个波动阈值 k。请找出最长的连续子数组（区间），使得该区间内最大值与最小值的差不超过 k。

方法一：朴素滑动窗口法 <pre> data = [75, 76, 76, 75, 78, 80, 95, 95, 95, 82, 81, 79] k = 5 ; left = 0 max_len = 0 ; start_index = 0 left = 0 for right in range(len(data)): while left <= right and max(data[left:right+1]) - min(data[left:right+1]) > k: left += 1 if right - left + 1 > max_len: max_len = right - left + 1 start_index = left print("最长平滑区间: 从第%d 个数开始, 长度: %d 个数" % (start_index+1, max_len)) </pre>	方法二：滑动窗口+单调队列优化法 <pre> data = [75, 76, 76, 75, 78, 80, 95, 95, 95, 82, 81, 79] k = 5 ; left = 0 max_len = 0 ; start_index = 0 max_q = [] ; min_q = [] for right in range(len(data)): while max_q and data[max_q[-1]] <= data[right]: max_q.pop() max_q.append(right) while min_q and data[min_q[-1]] >= data[right]: min_q.pop() min_q.append(right) while data[max_q[0]] - data[min_q[0]] > k: if max_q[0] == left: max_q.pop(0) if min_q[0] == left: min_q.pop(0) left = left + 1 current_len = right - left + 1 if current_len > max_len: max_len = current_len start_index = left print("最长平滑区间: 从第%d 个数开始, 长度: %d 个数" % (start_index+1, max_len)) </pre> 如果超过k差值，去掉一个最大值或最小值，但前提是最大值或最小值在区间的两边缘
--	--

四、真题回顾

【真题 202306T12】已排序的列表 a 有 n 个整型元素，现要查找出现次数最多的值并输出。若出现次数最多的值有多个，则输出最前面的一个。实现该功能的程序段如下，方框中应填入的正确代码为 **B**

c,m,v=1,1,0

for i in range(1, n):

print(a[v])

A. if a[i]==a[i-1]: <pre> c+=1 if c>m: m=c v=i else: c=1 </pre>	B. if a[i]==a[i-1]: <pre> c+=1 if c>m: m=c v=i else: c=1 </pre>	C. if a[i]==a[i-1]: <pre> c+=1 else: if c>m: m=c v=i-1 c=1 </pre>	D. if a[i]==a[i-1]: <pre> c+=1 else: if c>m: m=c v=i-1 c=1 </pre>
---	---	---	---

最后一段连续值未判断

【真题 202506T14】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：

(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（若仍有多个，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

i=0 ①

while i < len(data):

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

total=0

k=i

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

total+=data[i]

i+=1

n = i - k ②

if n>maxn:

maxn=n

start=k

maxt=total

elif n == maxn and total < maxt ③:

start=k

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

五、区间综合练习

在 T+1 交易制度下，投资者买入的股票不能在当天卖出，必须等到下一个交易日才能进行卖出操作。本程序用来模拟股票交易过程，功能是查询指定日期区间内某只股票的持有天数(卖出日期-买入日期+1)及其投资收益。细则如下：

①查询开始日期前股票数量为 0，结束日期后股票数量为 0

②股票交易不考虑佣金，印花税等所有费用

③买入部分的股票不能在当天卖出，同时委托的卖出数量不能超过持有数量，否则不合规

④每次交易后动态更新持有的股票价格及数量，且约定更新后股票价格不会出现负数值

⑤每只股票每天最多进行二次交易，且不能均为卖出操作

例如 st=[[1,2,'A',8,1000,1],[5,7,'A',10,1000,0]]，表示 A 股票在第 1 天 2 时段买入 1000 股，价格为 8 元，到第 5 天 7 时段全部卖出，持有 5 天，投资收益 2000 元。

(1) 若 s[[1,1,'A',8,1000,1],[3,5,'A',8.5,2000,0],[3,8,'A',7.9,2000,1],[8,2,'A',8.8,1000,0]]，A 股票的交易是否合规？B (单选，填字母：A.是 / B.否)

(2) 定义函数 stlnk(st)，功能是为每只股票添加指针链接，请在划线处填入合适代码。

def stlnk(st): head={}

将同类型的股票整理成一条链

n=len(st);pre={}

n=交易记录总数；pre存每只股票上一条记录的索引

for i in range(n):

st[i].append(-1)

给每条记录新增第7位（索引6），初始为-1（尾节点标记）

k=st[i][2]

k=当前记录的股票名称（如'A'/'B'）

if k not in head:

head[k]=i

若该股票首次出现，head指向当前记录（链表头）

else:

st[pre[k]][6]=i

若不是首次，上一条记录的第6位指向当前记录（链表链接）

pre[k]=i

更新pre：当前记录成为该股票的“上一条记录”

链队

(3)定义函数 ststat(stname)，功能是查询指定股票的持有天数及投资收益，查询结果如图所示，请在划线处填入合适代码。

```
def ststat(stname):
    q=head[stname]
    days=0 # 总持有天数
    dayst=st[q][0] # 初始持仓开始日期（首笔交易的日期）
    styield=0 # 总投资收益
    while q!=-1:
        price=st[q][3] # 初始化持仓价格（首笔必为买入，价格=首笔买入价）
        num=st[q][4] # 初始化持仓数量（首笔买入数量）
        while head[stname]!=-1: # 内层循环：处理当前持仓周期的交易
            p=q;q=st[q][6] # p=当前节点，q=下一个节点（链表后移）
            if q!=-1 and st[q][5]==1: # 下一条是买入交易
                price=(price*num+st[q][3]*st[q][4])/(num+st[q][4]) # 加权平均更新持仓价格
                num+=st[q][4] # 持仓数量累加新买入数量
            elif q!=-1 and st[q][0]>st[p][0]: # 当天卖出前无买入 # 下一条是卖出，且卖出日期>上一笔交易日期（T+1合规）
                if num-st[q][4]==0: # 卖出后持仓为0（清仓）
                    days+=st[q][0]-dayst+1 # 持有天数计算
                    styield+=st[q][3]*st[q][4]-price*num # 收益=卖出总金额 - 持仓总成本
                    head[stname]=-1 # 清空该股票的头节点（当前持仓周期结束）
                    q=st[q][6]
                elif num-st[q][4]>0: # 卖出后仍有持仓（部分卖出）
                    price=(price*num-st[q][3]*st[q][4])/(num-st[q][4]) # 更新持仓价格
                    num=num-st[q][4] # 持仓数量减少卖出数量
                else: # 卖出数量>持仓数量（不合规）
                    print(str(st[q][0])+"日"+str(st[q][1])+"时段卖出不合规")
            elif q!=-1: # 当天卖出前有买入 # 下一条是卖出，且卖出日期=上一笔交易日期
                if num-st[q][4]-st[p][4]>=0: # 在当天之前买入的量是否够卖出
                    price=(price*num-st[q][3]*st[q][4])/(num-st[q][4])
                    num=num-st[q][4]
                else:
                    print(str(st[q][0])+"日"+str(st[q][1])+"时段卖出不合规")
            else: # q== -1 无下一条记录，终止内层循环
                break
        if q!=-1: # 还有下一个持仓周期（清仓后又买入）
            head[stname]=q # 更新链表头为新的起始节点
            dayst=st[q][0] # 更新持仓开始日期为新买入日期
    return days,styield
```

请输入要查询的股票名称:A
2 日 8 时段卖出不合规
持有天数: 7 投资收益: 5600 元

读取股票交易记录，存入列表 st，并按交易时间升序排序，代码略。st 中每个元素包含 6 个数据项：交易日期，交易时段，股票名称，交易价格，交易数量和买卖标识(1,0 分别表示买,卖)，如 s=[[1,3,'B',15,2000,1],[1,7,'A',8,3000,1],[2,1,'A',7.8,2000,1],[2,8,'A',8.5,6000,0],[3,4,'A',8.5,5000,0],[3,6,'B',14.5,2000,0],[5,7,'A',7.9,3000,1],[8,1,'A',8.8,3000,0]]"

```
head={}
stlnk(st)
stname=input("请输入要查询的股票名称: ")
dur,styield=ststat(stname)
print("持有天数: ",dur,"投资收益: ",int(styield),"元")
```