20241128题解

A（P6360）

如果我们暴力处理每两个区间的距离，则时间复杂度是O(n^2l)的，会超时。

空间限制如果有问题的话可以开离散化来存。

对于题目中的定义，我们发现，计算[x1,x1+l-1]到[x2,x2+l-1]的距离，和计算[x1+1,x1+l]到[x2+1,x2+l]的距离时，两者的变化量是相同的，因此可以把左端点距离差相同的区间的距离一起计算。所以我们可以枚举x1，枚举差，然后就可以O(n^2)计算出所有对区间的距离了。

另外如果空间不够，可以在存储的时候离散化一下。

B（P8165）

对于操作1，由于K小，直接暴力修改即可。

对于操作2，连续子序列元素和可以用前缀和来快速计算，例如a[l]+a[l+1]+...+a[l+m-1]=S[l+m-1]-S[l-1]。转化为前缀和的式子可以写成S[l+m-1]+S[l+m]+...+S[r]-(S[l-1]+S[l]+...+S[r-m])。因此可以用一个支持区间查询的数据结构维护前缀和即可。

这样的话我们重新看回操作1，将x替换为y可以看成x这个位置加上y-x，如果是a[i]加上x的话，S[i]到S[n]都会加x，所以可以将K个单点修改转化为K个区间修改，因此需要一个支持区间修改区间查询的数据结构，直接线段树就行了。

当然也有别的做法，还有一种维护i\*a[i]的区间和的做法。（此时将式子转化为1\*a[l]+2\*a[l+1]+...+2\*a[r-1]+1\*a[r]，中间可能有一段的系数都是m，因此可以把求和拆成3段，发现要维护a[i]和i\*a[i]的区间和，所以还是线段树）

C（P10849）

纯粹的贪心，不一定有前两题难（虽然是蓝的）。

首先肯定最大化t<s的数量，然后尽可能保留t-s的匹配即可。我们从小到大枚举s，每个s在[0,s)范围内尽可能多去找t，由于匹配区间内最大的t肯定更优，所以我们可以用一个栈来维护这个过程，然后就没了。

这个栈具体维护的是没有分配完的房间等级，房间数量如果归零合计就从栈顶弹出，当等级s的人分配完之后就将等级s的房间入栈；如果分配完之前栈空的话剩下的人肯定是点差评的，在答案里减掉之后当成分配完即可。