golem

如果数列的总和是 sum，那么就是要找一个和最大并且小于 $sum-k$ 的子段。

对于一个前缀和为 now 的前缀。要找到在他之前的前缀和 $\geq now-sum+k$ 的最大前缀和，用 set 查后继即可。

fire

考虑对于每个员工找出他在哪一天加入和被裁，那么我们就可以在这两天之间加入一条边，意思是这两天的人事不能相同。然后就变成了图染色问题。

公司人员进出遵循先进先出的原则，因此公司中的员工显然可以用一个栈来表示。本质不同的边只有 $O(n)$ 条。

首先能用一或二染色的可以直接特判。下面我们证明这个图一定能被三染色。

这个图有什么特殊性质呢？还是栈的结构。由于先进先出的性质，所以一定不存在边 $(x_1,y_1),(x_2,y_2)$ 使得 $x_1<x_2,y_1<y_2$。

归纳地构造。假设一个图的两边的点 x,y 的颜色已经被确定了，中间的点还不知道。我们选一个点 z 使得不存在边 (x,x_2) 使得 $x_2>z$ 和边 (y_2,y) 使得 $y_2<z$。由于上面图的性质这样的点一定存在。那么把 z 染成和 x,y 不一样的那个颜色，然后把图分成两个部分就可以了。看不懂可以画图理解一下。

gene

题意就是可以单点加，超集加，子集加，询问子集和。

先规定 $|S|$ 是 S 二进制下 1 的个数。

设 S 拆成前 $\dfrac{n}{2}$ 位和后 $\dfrac{n}{2}$ 位后是 S_1,S_2，

对于单点加，有一个经典的折半：设 $g_{i,j}$ 表示 $S_1=i$，$S_2\subseteq j$ 的 f_S 之和。修改的时候只需要改 $g_{S_1,*}$，查询的时候查询 g_{*,S_2}，枚举量都是 $2^{\frac{n}{2}}$ 的。

对于超集加和子集加，我们可以考虑修改对询问的贡献，如果给 S 的超集加 1，查询 T 的子集和，那么贡献是 $[S\subseteq T]2^{|T|-|S|}$。这个根据定义就能得出来。发现和单点加是差不多的，单点加的贡献是 $[S\subseteq T]$，相当于说我们再开一个数组维护超集加，只需要和单点加一样做，只是加的时候 $c:=c*2^{-|S|}$，查的时候把答案乘上 $2^{|T|}$ 就可以。

对于子集加修改对询问的贡献是 $2^{|S\&T|}$。这个可以类似地维护。那么 $2^{|S\&T|}=2^{|S_1\&T_1|}2^{|S_2\&T_2|}$。我们维护 $f_{i,j}$ 表示 $\sum\limits_{S_1=i}2^{j\&S2}$，发现修改只需要修改 $f_{S_1,*}$，查询时答案就是 $\sum\limits_{i}2^{T_1\&i}f_{i,T_2}$。复杂度就是 $2^{\frac{n}{2}}$。

应该还可以根号重构做到 $q2^{\frac{n}{2}}\sqrt{n}$，不知道能不能过。

contest

既有取 $\min$ 又有取 $\max$ 实在是太麻烦了，我们先把所有取 $\min$ 操作拿出来，设其中最小的那一个为 x。那么首先原序列应对 x 取 $\min$，$\leq x$ 的取 $\max$ 操作没有用（一定会被原样执行）。这样就变成了所有操作的参数全部 $>x$，然而原序列全部 $<x$，所以原序列没有用。

进一步地，我们将取 $\min$ 操作从小到大排序。取 $\max$ 操作都可以放在任意对 c 取 $\min$ 操作的前面，可以看作对 c 取 $\max$。这就相当于我们扔掉所有取 $\min$ 操作，然后取 $\max$ 可以对原来的这个值，或者任意比它小的取 $\min$ 操作的值取 $\max$。

于是记 $f(l,r,k)$ 表示通过取 $\max$ 操作将 $[l,r]$ 这个区间全部变成 $\geq k$ 的方案数（序列种类数）。这样只有 $[l,r]$ 以内的取 $\max$ 操作才有用，从大到小枚举 k 转移即可，具体地，固定 l,k 扫 r，维护 g_i 表示用 $[l,r]$ 中的操作把 $[l,i]$ 变成 $\geq k$ 的方案数，维护每个位置可不可以通过取 $\max$ 变成 k，称作是否被激活，转移就是：

$g_{i-1} * f_{i+1,j,k+1} \to g_j$（$i$ 被激活）
$g_r \to f_{l,r,k}$（处理完右端点 $\leq r$ 的区间后）

直接转移有 $O(n^5)$，预处理一些东西就可以有 $O(n^4)$。