Solution

Larunatrecy

目录

A. 重建：地下铁道 (railway)

题目大意

给定一个 n 个点的环，结点 i 和 $i+1$ 之间有长度为 l_i 的无向边，你可以给每条边任意定向。有 m 个任务，每个任务要从 s_i 沿着边的方向走到 t_i，那么代价是 $w_i \times L$，其中 L 是走的长度。需要最小化总代价。

$n, m \leq 5 \times 10^5$。

吐槽环节

$n, m \leq 20$

枚举每条边的方向或者枚举每次任务是顺时针还是逆时针，复杂度是 $O(2^n(n+m))$ 或者 $O(2^m(n+m))$。
期望得分：15。

$m \leq 18$

枚举每次任务是顺时针还是逆时针，然后用你喜欢的东西维护一下。复杂度 $O(2^m \log n)$。
期望得分：30。

正解一

我们称一个路线经过了 $(1, n)$ 的任务是从外部走的，否则是从内部走到。考虑一个性质，如果至少存在一个 $s_i < t_i$ 是从外部走的，并且至少存在一个 $s_i > t_i$ 的任务是从外部走的，那么一定不合法。

故下面条件至少满足一个：

- 所有 $s_i < t_i$ 的任务都是从内部走的。
- 所有 $s_i > t_i$ 的任务都是从内部走的。

这就是说两种任务中有一种的所有方向都是确定的，我们可以枚举是哪一种（不妨认为是 $s_i < t_i$ 的）。

然后有些边被定向了，而如果有一个 $s_i > t_i$ 的任务覆盖了被定向的边，它的方向就也被确定了，迭代下去可以确定所有被确定方向的边和任务。

迭代的过程可以线段树之类的维护一下。

正解一

考虑如果一个 $s_i > t_i$ 的任务从外部走了，那么所有不被 $[t_i, s_i]$ 包含的 $[t_j, s_j]$ 都也要从外部走。换句话说最终从内部走的任务一定是某个子 $[t_i, s_i]$ 内的所有区间。

这是一个二维矩形和，扫描线求一下即可，复杂度是 $O(m \log n)$。

这个做法要写好多数据结构，太麻烦了！

正解二

考虑最终从内部走的 $s_i > t_i$ 的任务，它一定是把区间按照长度从小到大排序后的一个前缀。

证明很简单，一个从外部走的任务最多只能提供长度大小条可以被反向定向的空间。

判断一个前缀是否合法也很简单，求出后缀的所有区间的交，维护一下前缀内 l_i 的最小值和 r_i 的最大值，判断一下是否包含即可。

只需要对所有区间排序，复杂度也是 $O(m \log m)$。

B. 走过安眠地的花丛 (death)

题目大意

构造一个 n 个点有边权的 DAG，使得所有 $1 \sim n$ 的路径的长度均在一个给定的列表 $a_1, a_2 \ldots a_k$ 中，并且所有列表中的数字均可以表示成某条路径的长度。
要求 n 尽可能小。

吐槽环节

$n = 92$

设 $V = 2000, b = \sqrt{V}, c = \lfloor \frac{V}{b} \rfloor$

- $\forall i = 0, 2 \ldots c$ 从 1 向 $2 + i$ 连权值为 0 的边
- $\forall i = 0, 2 \ldots b - 1$ 从 $c + 3 + i$ 向 n 连权值为 0 的边 $\forall i \in A$, 从 $2 + (i/b)$ 向 $c + 3 + (i \bmod b)$ 连权值为 i 的边。

大致需要 $1 + 44 + 46 + 1 = 92$ 个点。

$n = 69$

每两个元素分成一组，每组内有 $(0,0),(1,0),(0,1),(1,1)$ 四种情况，其中 $(0,0)$ 是不需要考虑的。
我们考虑将原本的一个源点变成 $1,2$ 两个源点，从 1 向 2 连权值为 1 的边，这样我们就可以通过和这两个原点的 4 种不同连边状态区分每一组。
每组内还是按照 $0,1,\ldots s_i/b$ 分组，其中 s_i 是某一组的个数，然后统一连到 $0,1\ldots b-1$ 的点上，最后连到汇点。
这样做别忘了每一组内点数是上取整，所以最后每组会多一个。
这样的点数约为 $2+(1000/b)+3+b+1$，取 $b=31$ 的话 $n=69$。

$n = 62$

仿照上面的思路，我们可以每 $3, 4, 5$ 个分成一组。
3 个时的代价是 $3 + 26 + 7 + 25 + 1 = 62$ 个。
4 个时的代价是 $4 + 22 + 15 + 22 + 1 = 64$ 个。
5 个时的代价是 $5 + 20 + 31 + 20 + 1 = 77$ 个。
更大的时候只会更劣。

$n = 57$

以 4 个分组为例，实际有用的状态其实只有 8 个，这是因为如果某一组的第一个数是 0，我们可以直接跳过这个位置从下一个非零位置开始划分，这样只有 2^3 种状态。

此时状态数为 $4 + 22 + 8 + 22 + 1 = 57$ 个。

如果按照 3 个分组的话，状态数为 $4 + 26 + 4 + 25 + 1 = 59$ 个，还不如 4 个划分了。

$n = 56$

把汇点删了，用最后一层的最后一个作为汇点即可。

$n = 53$

事实上 $1, 2, 3, 4$ 四个点也可以删成一个。

把 2^3 个组里每个组的第一个作为这个组的代表元素，然后代表元素向同组内的其他元素连 0 的边，这样的话只需要使得代表元素之间符合要求即可。而这是简单的，从 1 向第 $0, 1, 2, 4$ 组的代表元分别连 $0, 1, 2, 3$ 的边，然后每个代表元向其超集代表元连 0 边即可。即使某个组是空的也无所谓，直接新建就行，这个地方不影响点数。

最后是 53 个点。

C. 昔在、今在、永在的题目 (problem)

题目大意

维护一个长度为 n，值域为 m 的序列 a，q 次修改，支持单点修改，每次修改后查询如果在 $i < j, a_j = a_i \bmod m + 1$ 的点对之间连边后的最大匹配数量。

$1 \leq n, m, q \leq 3 \times 10^5$。

吐槽环节

$n, m, q \leq 300$

设 a_i 为值 $= i$ 的个数，b_i 为在 i 和 $i \bmod m + 1$ 之间最多能选出多少匹配，最终在 i 和 $i \bmod m + 1$ 之间选出了 x_i 对匹配。
那么我们要满足：

- $x_i \leq b_i$
- $x_i + x_{i \bmod m+1} \leq a_{i \bmod m+1}$

容易说明符合这个式子的都是可以取到的，具体证明可以考虑在 i 和 $i \bmod m + 1$ 匹配时用 i 的前 x_i 个和 $i \bmod m + 1$ 的后 x_i 个，这样一定是最优的。
然后直接枚举 x_1，然后每次贪心地令 $x_{i+1} = \min(b_i - x_i, c_{i+1})$，容易证明这样子的正确性。
复杂度 $O(mnq)$。

正解一

设 $f_i(X)$ 表示 $x_i = X$ 时 x_m 等于什么，$g_i(X)$ 表示 $\sum_{j=i}^{n} x_j$，可以证明 f, g 都是 $O(1)$ 段分段一次函数，直接用线段树维护即可。维护 b_i 可能需要平衡树或者动态开点线段树来维护匹配。复杂度 $O((n+q)\log n)$

正解二

直接把线性规划对偶，可以得到下面的问题：

$$\min \sum_{i=1}^{m} a_i y_i + \max(0, 1 - y_i - y_{i+1}) b_i \tag{1}$$

$$y_i \leq 1 \tag{2}$$

可以证明最优解中 y_i 一定会取到 $0, 1, \frac{1}{2}$，因此使用线段树维护矩阵乘法即可。
复杂度 $O((n+q)\log n)$。

Thanks!