

通信 解题报告

成都七中 王修涵

1 题目来源

ONTAK 2018 Nadajniki¹。

2 题目大意

在数轴上有 n 个信号发射器，第 i 个信号发射器在 a_i 位置，每个信号发射器可以调整它的发射半径为 r_i 。定义 j 在 i 的发射范围内当且仅当 $|a_i - a_j| \leq r_i$ 。如果两个点互相在发射范围内，它们可以互相通信。定义一种方案代价是有序数对 (a, b) 的数量，满足 b 在 a 的发射范围内。求一种最小代价的方案，使得任意两个信号发射器可以直接或间接通信。

3 子任务

对于100%的数据，满足 $2 \leq n \leq 300, 0 \leq a_i \leq 2 \times 10^{18}$ ，读入的 a_i 单调递增。

编号	分值	$n \leq$	特殊性质
1	10	8	
2	10	12	
3	10	60	$a_i = 2^i$
4	20	40	
5	30	100	
6	20	300	

¹题目链接：https://sio2.mimuw.edu.pl/c/wiekuisty_ontak2018/p/nad/

4 暴力算法

4.1 算法一

我们可以先选择一棵生成树，再根据树边确定 r_i 并计算代价。

使用prufer编码枚举所有的生成树，时间复杂度 $O(n^n)$ ，期望得分10分。

4.2 算法二

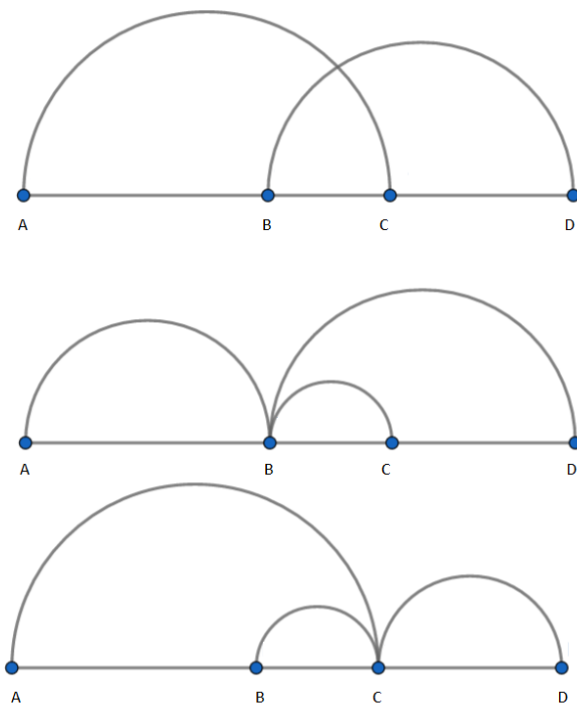
考虑状压DP，记 $f(S, root, range)$ 表示一棵子树（不包含根）的最小代价，其中 S 是子树的所有点的集合， $root$ 是子树的根， $range$ 是 $root$ 只考虑儿子的限制时 $root$ 的发射范围。

转移时枚举一棵子树接上去，可以用前缀和等方式优化，时间复杂度 $O(3^n poly(n))$ ，期望得分20分。

5 发掘性质

5.1 性质一

引理5.1. 存在一个最优解，满足在其生成树上，对于任意 $A < B < C < D$ ，边 (A, C) 和边 (B, D) 不同时存在。



Proof. 如上图，第一种方案是一种存在边 (A, C) 和 (B, D) 的解，当 $|AB| \leq |CD|$ 时，我们调整为第二种方案，否则我们调整为第三种方案。

不难发现，新的方案不但没有改变连通性，还有使得 A, B, C, D 的发射范围不变或者变得更小。 \square

根据上述引理，我们不难想到用区间DP刻画这个模型。为了方便描述，记 $\text{cover}(v, r_v)$ 表示将 v 的发射半径设定为 r_v 时， v 发射范围内的点数。

5.2 算法三

记 $f(s, t, r_s, r_t)$ 表示 (s, t) 已经连通， s, t 的发射半径分别为 r_s, r_t 的情况下，将 $[s, t]$ 连成一个连通子图， $s, s+1, \dots, t$ 产生的最小代价，有转移：

$$f(s, t, r_s, r_t) = \begin{cases} 0 & t = s + 1 \\ \min_{s < m < t} f(s, m, r_s, r_m) + f(m, t, r_m, r_t) + \text{cover}(m, r_m) & t \neq s + 1 \end{cases}$$

这里的 m 和 r_m 需要满足 m 能与 s 或 t 连边。

第一条转移是显然的，第二条转移的意思是，区间 (s, t) 内至少有一个点与 s 或者 t 相连，枚举这个点 m 和它的发射范围 r_m ，变成两个子问题递归处理。

再记 $g(s, t, r_s, r_t)$ 表示 (s, t) 还没有连通时的最小代价，有转移：

$$g(s, t, r_s, r_t) = \min_{s < m < t} f(s, m, r_s, r_m) + g(m, t, r_m, r_t) + \text{cover}(m, r_m)$$

这里的 m 和 r_m 需要满足 m 能与 s 连边。

还有一种转移是，如果 s 能与 t 连边，可以用 $f(s, t, r_s, r_t)$ 更新 $g(s, t, r_s, r_t)$ 。

统计答案时，枚举 r_0 和 r_{n-1} 用 g 数组统计答案即可。

状态数为 $O(n^4)$ ，转移数为 $O(n^2)$ ，总时间复杂度 $O(n^6)$ ，期望得分50分。

6 算法优化

6.1 性质二

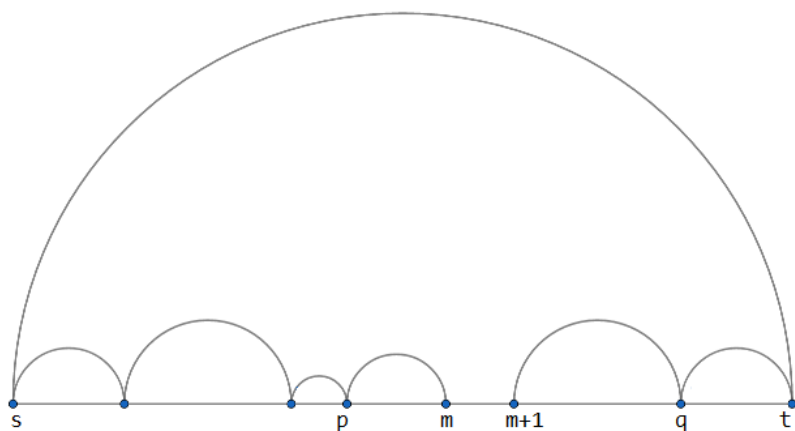
引理6.1. 在满足上一个性质的解中，对于任意一条边 (s, t) ，存在 $s \leq m < t$ 满足对于任意一条在 (s, t) 内部的边 (l, r) ， $l \leq m$ 和 $r > m$ 不同时成立。

Proof. 考虑反证法，即 (s, t) 内部的边跨过了所有间隔。因为没有两条边相交，那么所有极长的边一定是 $(s, p_1), (p_1, p_2), \dots, (p_k, t)$ 的形式。

因为 s 和 t 已经有一条边了，所以产生了环，矛盾。 \square

6.2 算法四

回到算法三，不难发现对于每个状态记录 r_s 和 r_t 是十分浪费的。考虑重新定义状态，记 $f(s, t)$ 表示 (s, t) 已经有边了，将 (s, t) 内部连通， $s+1, s+2, \dots, t-1$ 产生的最小代价。根据上述引理， (s, t) 内部的边一定是下图所示的形式：



为了辅助DP, 记 $gl(s, m)$ 表示将 (s, m) 用若干条极长边连通, $s+1, s+2, \dots, m$ 产生的最小代价, 记 $gr(m+1, t)$ 表示将 $(m+1, t)$ 用若干条极长边连通, $m+1, m+2, \dots, t-1$ 产生的最小代价。

再记 $fl(s, p, m)$ 表示将 (s, m) 用若干条极长边连通, 最后一条边是 (p, m) 时, $s+1, s+2, \dots, m-1$ 产生的最小代价, 记 $fr(m+1, q, t)$ 表示将 $(m+1, t)$ 用若干条极长边连通, 第一条边是 $(m+1, q)$ 时, $m+2, m+3, \dots, t-1$ 产生的最小代价。

考虑转移:

对于 $f(s, t)$, 我们枚举 m , 有转移

$$f(s, t) = \begin{cases} 0 & t \leq s+1 \\ \min_{s \leq m < t} \{gl(s, m) + gr(m+1, t)\} & t > s+1 \end{cases}$$

对于 $gl(s, m)$ 和 $gr(m+1, t)$, 我们枚举 p 和 q , 有转移

$$gl(s, m) = \min_{s \leq p < m} \{fl(s, p, m) + \text{cover}(m, a_m - a_p)\}$$

$$gr(m+1, t) = \min_{m+1 < q \leq t} \{fr(m+1, q, t) + \text{cover}(m+1, a_q - a_{m+1})\}$$

由于 fr 和 fl 转移类似, 所以这里只考虑 fl 的转移:

$$fl(s, p, m) = \min_{s \leq q < p} \{fl(s, q, p) + f(p, m) + \text{cover}(p, \max(a_p - a_q, a_m - a_p))\}$$

统计答案还需要一个DP：记 $g(l, r)$ 表示已经将 r 之前的所有点连通，与 r 相连的最左边的点是 l 时， r 之前的所有点产生的所有代价，有转移：

$$g(l, r) = \begin{cases} f(1, r) + \text{cover}(1, a_r - a_1) & l = 1 \\ \min_{p < l} \{g(p, l) + f(l, r) + \text{cover}(l, \max(a_l - a_p, a_r - a_l))\} & l > 1 \end{cases}$$

答案是 $\min_{i < n} \{g(i, n) + \text{cover}(n, a_n - a_i)\}$ 。

状态数 $O(n^3)$ ，转移 $O(n)$ ，时间复杂度 $O(n^4)$ ，期望得分80分。

对于这个子任务，还有一些其他的 $O(n^4)$ 的做法，但由于思想类似且优化比较困难或者不能优化，故不再赘述。

6.3 算法五

在算法四中，瓶颈在于 $fl(s, p, m)$ 的转移，考虑优化这一部分。

注意到 $\max(a_p - a_q, a_m - a_p)$ 存在一个分界点，满足在这个分界点前面的 q ，这个值为 $a_p - a_q$ ，在分界点后面的 q ，这个值为 $a_m - a_p$ 。

对于第一种情况，我们维护 $fl(s, q, p) + \text{cover}(p, a_p - a_q)$ 的前缀最大值；对于第二种情况，我们维护 $fl(s, q, p)$ 的后缀最大值，就可以做到 $O(1)$ 转移了。

时间复杂度 $O(n^3)$ ，期望得分100分。

7 另一个部分分

对于 $a_i = 2^i$ 的数据，有更强的性质：

引理7.1. 对于算法四中的 $f(s, t)$ ，最优的连边方法一定是对于 $s < i < t$ ， i 向 s 连边。

Proof. 由于 $a_i = 2^i$ ，所以一个点如果向右连边，它一定能覆盖左边的所有点，在 s 和 t 已经连通的情况下，内部的点向右连边不如向左连边。□

7.1 算法六

对于这部分，可以 $O(1)$ 计算算法四中的 $f(l, r)$ ，时间复杂度 $O(n^3)$ ，期望得分10分，与其他算法结合可以获得更高的分数。

7.2 一些拓展

事实上，这部分可以做一个更大的范围： $n \leq 10^6$ 。这是原题的第三个子任务。

假设 g 的转移分成了 k 段，记第 i 段转移的长度是 b_i ，那么对于第 i 个向右连的点，它的代价是 $\sum_{j \leq i} b_j$ ，而第 i 段内部的点的代价之和是 $\frac{b_i(b_i-1)}{2}$ 。

所以代价之和：

$$\begin{aligned} ans &= \sum_{i=1}^k \sum_{j \leq i} b_j + \sum_{i=1}^k \frac{b_i(b_i-1)}{2} \\ &= \frac{1}{2} \left(\sum_{i=1}^k b_i^2 - (2k-2i+1)b_i \right) \\ &= \frac{1}{2} \left((2k+1)n + \sum_{i=1}^k b_i^2 - 2ib_i \right) \\ &= \frac{1}{2} \left((2k+1)n + \sum_{i=1}^k (b_i-i)^2 - \sum_{i=1}^k i^2 \right) \end{aligned}$$

所以只需要最小化 $\sum_{i=1}^k (b_i-i)^2$ 即可。

因为 $\sum_{i=1}^k (b_i-i)$ 是固定的，根据均值不等式，当 b_i-i 尽量平均时，这个值最小。

而当最优解中出现 $b_i < 0$ 的时候，答案显然会更劣，所以可以忽略这些情况。枚举 k 之后，可以 $O(1)$ 计算最优解。

而事实上这一部分的答案也存在规律：差分之后的序列是3出现2次，4出现3次，5出现4次，以此类推。通过打表找规律也可以获得这部分的分数。

8 题目总结

对于前两个子任务，思维难度较低，但考虑到考场上选手的策略等问题，预计有80%的选手能够通过。

对于第三个子任务，可以通过打表找规律或者发现性质后设计一个简单的动态规划来解决，预计有70%的选手能够通过。

对于第四个子任务，需要挖掘本题最关键的性质，思维难度有一定提升。一些较优随机化算法可能通过这个子任务，预计有50%的选手能够解决。

对于第五个和第六个子任务，考察了选手设计动态规划状态并分析出转移的能力，思维难度较高，预计有30%的选手能过解决。

总的来说，本题是一道动态规划题目，涉及到生成树，状压DP，区间DP，公式推导等知识点，思维难度较高，代码难度较低，不仅考察了选手动态规划的基本功，还考察了选手观察题目，分析性质的能力，难度介于近年NOI第二题和第三题之间。

参考文献

- [1] Haisheng Tan, Tiancheng Lou, Francis C.M. Lau, Yuexuan Wang, and Shiteng Chen "Minimizing Interference for the Highway Model in Wireless Ad-Hoc and Sensor Networks"
- [2] Tiancheng Lou, Haisheng Tan, Yuexuan Wang, and Francis C.M. Lau "Minimizing Average Interference through Topology Control"