Codeforces水题选讲

infinityedge

山东省莒县第一中学

March 12, 2017

题目介绍

- 题目选自去年8月的两场CF，非常的水（不过那时我不会做，只能做Div2的A题）

题目介绍

- 题目选自去年8月的两场CF，非常的水（不过那时我不会做，只能做Div2的A题）
- 难度大约等同于「麻将」。

题目介绍

- 题目选自去年8月的两场CF，非常的水（不过那时我不会做，只能做Div2的A题）
- 难度大约等同于「麻将」。
- 最后一个讲题，讲这些水题给各位秒了一晚上题的dalao能休闲一下。

Outline

题目描述

- 给定 N个点，其中第 i 个点的坐标为 x_i。

题目描述

- 给定 N个点，其中第 i 个点的坐标为 x_i。
- 每个点 i 还有四个参数 a_i, b_i, c_i, d_i

题目描述

- 给定 N个点，其中第 i 个点的坐标为 x_i。
- 每个点 i 还有四个参数 a_i, b_i, c_i, d_i
- 从第i个点走向第j个点需要的时间为：

题目描述

- 给定 N个点，其中第 i 个点的坐标为 x_i。
- 每个点 i 还有四个参数 a_i, b_i, c_i, d_i
- 从第i个点走向第j个点需要的时间为：
- $|x_i - x_j| + c_i + b_j, j < i$

题目描述

- 给定 N个点，其中第 i 个点的坐标为 x_i。
- 每个点 i 还有四个参数 a_i, b_i, c_i, d_i
- 从第i个点走向第j个点需要的时间为：
- $|x_i - x_j| + c_i + b_j, j < i$
- $|x_i - x_j| + d_i + a_j, j > i$

题目描述

- 给出起点 S 和终点 T ，要求从S出发，必须经过每一个点一次，最后走到 T 结束，要求总用时最短。

题目描述

- 给出起点 S 和终点 T ，要求从S出发，必须经过每一个点一次，最后走到 T 结束，要求总用时最短。
- $1 \le x_1 \le x_2 .... x_n \le 10^9$

题目描述

- 给出起点 S 和终点 T ，要求从S出发，必须经过每一个点一次，最后走到 T 结束，要求总用时最短。
- $1 \leq x_1 \leq x_2....x_n \leq 10^9$
- $1 \leq a_i, b_i, c_i, d_i \leq 10^9$

题目描述

- 给出起点 S 和终点 T ，要求从S出发，必须经过每一个点一次，最后走到 T 结束，要求总用时最短。
- $1 \leq x_1 \leq x_2 \ldots x_n \leq 10^9$
- $1 \leq a_i, b_i, c_i, d_i \leq 10^9$
- Step 1：对于 100% 的数据，$1 \leq n \leq 5000$(原题)

题目描述

- 给出起点 S 和终点 T ，要求从S出发，必须经过每一个点一次，最后走到 T 结束，要求总用时最短。
- $1 \leq x_1 \leq x_2....x_n \leq 10^9$
- $1 \leq a_i, b_i, c_i, d_i \leq 10^9$
- Step 1：对于 100% 的数据，$1 \leq n \leq 5000$(原题)
- Step 2：对于 200% 的数据，$1 \leq n \leq 10^5$

题目描述

- 给出起点 S 和终点 T ，要求从S出发，必须经过每一个点一次，最后走到 T 结束，要求总用时最短。
- $1 \le x_1 \le x_2 \ldots x_n \le 10^9$
- $1 \le a_i, b_i, c_i, d_i \le 10^9$
- Step 1：对于 100% 的数据，$1 \le n \le 5000$(原题)
- Step 2：对于 200% 的数据，$1 \le n \le 10^5$
- 我们来钦点一位同学来讲一下（不限数据范围）。

Step 1

- 本题做法很多，可贪心，可DP，可费用流

Step 1

- 本题做法很多，可贪心，可DP，可费用流
- 这里介绍贪心的做法。(费用流解法我不会)

贪心解法

- 本题的最短路不是传统意义上的最短路，路径使一个序列，每个点对答案的贡献只与路径序列中与该点相邻的点有关。

贪心解法

- 本题的最短路不是传统意义上的最短路，路径使一个序列，每个点对答案的贡献只与路径序列中与该点相邻的点有关。
- 我们首先可以在路径序列先固定好 S 点和 T 点

贪心解法

- 本题的最短路不是传统意义上的最短路，路径使一个序列，每个点对答案的贡献只与路径序列中与该点相邻的点有关。
- 我们首先可以在路径序列先固定好 S 点和 T 点
- 剩余的点按照 i 递增的顺序依次插入到路径序列中，每次枚举插入的位置即可。

贪心解法

- 本题的最短路不是传统意义上的最短路，路径使一个序列，每个点对答案的贡献只与路径序列中与该点相邻的点有关。
- 我们首先可以在路径序列先固定好 S 点和 T 点
- 剩余的点按照 i 递增的顺序依次插入到路径序列中，每次枚举插入的位置即可。
- 复杂度$O(n^2)$，足以通过原题

贪心解法

- 本题的最短路不是传统意义上的最短路，路径使一个序列，每个点对答案的贡献只与路径序列中与该点相邻的点有关。
- 我们首先可以在路径序列先固定好 S 点和 T 点
- 剩余的点按照 i 递增的顺序依次插入到路径序列中，每次枚举插入的位置即可。
- 复杂度$O(n^2)$，足以通过原题
- 当时在CF比赛中写出正解的Div1+Div2总共只有不到50人，目前也仅有300余人AC。

Step 2

- 看来Step 1也是很好做的

Step 2

- 看来Step 1也是很好做的
- 那么Step 2呢？

Step 2

- 因为我们每次都是按照坐标从小到大的点贪心的

Step 2

- 因为我们每次都是按照坐标从小到大的点贪心的
- 所以转移时除了插入在S后面或T前面时，该点的坐标一定大于前后的两个点的坐标

Step 2

- 因为我们每次都是按照坐标从小到大的点贪心的
- 所以转移时除了插入在S后面或T前面时，该点的坐标一定大于前后的两个点的坐标
- 设当前的点为k，则k插入到i与j中间的代价为：

Step 2

- 因为我们每次都是按照坐标从小到大的点贪心的
- 所以转移时除了插入在S后面或T前面时，该点的坐标一定大于前后的两个点的坐标
- 设当前的点为k，则k插入到i与j中间的代价为：
- $$x_k - x_i + d_i + a_k + x_k - x_j + c_k + b_j - x_i + xj - c_i - b_j, i > j$$

Step 2

- 因为我们每次都是按照坐标从小到大的点贪心的
- 所以转移时除了插入在S后面或T前面时，该点的坐标一定大于前后的两个点的坐标
- 设当前的点为k，则k插入到i与j中间的代价为：
- $$x_k - x_i + d_i + a_k + x_k - x_j + c_k + b_j - x_i + xj - c_i - b_j, i > j$$
- $$x_k - x_i + d_i + a_k + x_k - x_j + c_k + b_j - x_j + xi - d_i - a_j, i < j$$

Step 2

- 由于$x_k + x_k + a_k + c_k$是固定的，我们可以算出其余最小值后加上该值，化简后为：

Step 2

- 由于$x_k + x_k + a_k + c_k$是固定的，我们可以算出其余最小值后加上该值，化简后为：
- $$-x_i + d_i - x_i - c_i, i > j$$

Step 2

- 由于$x_k + x_k + a_k + c_k$是固定的，我们可以算出其余最小值后加上该值，化简后为：
- $$-x_i + d_i - x_i - c_i, i > j$$
- $$-x_j + b_j - x_j - a_j, i < j$$

Step 2

- 由于$x_k + x_k + a_k + c_k$是固定的，我们可以算出其余最小值后加上该值，化简后为：
- $$-x_i + d_i - x_i - c_i, i > j$$
- $$-x_j + b_j - x_j - a_j, i < j$$
- 只有关于i和j中最大值的式子了，这样是不是变得十分美妙了？

Step 2

- 由于$x_k + x_k + a_k + c_k$是固定的，我们可以算出其余最小值后加上该值，化简后为：
- $$-x_i + d_i - x_i - c_i, i > j$$
- $$-x_j + b_j - x_j - a_j, i < j$$
- 只有关于i和j中最大值的式子了，这样是不是变得十分美妙了？
- 不错的，插入k之后，k就成为了k左右两条路径的最大值。

Step 2

- 化简之后，我们就可以用一个堆来维护插入每一个数之后在每一段路插入的代价.

Step 2

- 化简之后，我们就可以用一个堆来维护插入每一个数之后在每一段路插入的代价.
- 我们每次从heap里面pop出一个最小值，加上之前一堆关于k的式子，就是插入k之后总时间增加的值。

Step 2

- 化简之后，我们就可以用一个堆来维护插入每一个数之后在每一段路插入的代价.
- 我们每次从heap里面pop出一个最小值，加上之前一堆关于k的式子，就是插入k之后总时间增加的值。
- 插入k后，我们把$i->k$的路的代价$-x_k+d_k-x_k-c_k$和$k->j$的路的代价$-x_k+b_k-x_k-a_k$插入到堆里就好了啦

Step 2

- 化简之后，我们就可以用一个堆来维护插入每一个数之后在每一段路插入的代价.
- 我们每次从heap里面pop出一个最小值，加上之前一堆关于k的式子，就是插入k之后总时间增加的值。
- 插入k后，我们把$i->k$的路的代价$-x_k+d_k-x_k-c_k$和$k->j$的路的代价$-x_k+b_k-x_k-a_k$插入到堆里就好了啦
- 那么路径的起点和终点怎么处理呢？

Step 2

- 化简之后，我们就可以用一个堆来维护插入每一个数之后在每一段路插入的代价.
- 我们每次从heap里面pop出一个最小值，加上之前一堆关于k的式子，就是插入k之后总时间增加的值。
- 插入k后，我们把$i->k$的路的代价$-x_k+d_k-x_k-c_k$和$k->j$的路的代价$-x_k+b_k-x_k-a_k$插入到堆里就好了啦
- 那么路径的起点和终点怎么处理呢？
- 特判一下就好了啦qaq

Step 2

- 算法复杂度$O(nlog_2 n)$

Step 2

- 算法复杂度$O(nlog_2 n)$
- 可以通过Step 2 的加强数据。

Step 2

- 算法复杂度$O(nlog_2 n)$
- 可以通过Step 2 的加强数据。
- 启示：不能仅满足于std，要寻找更优的解法

Outline

题目描述

- 一年中有 2^n 天，有 k 个人，求至少两个人同一天生日的概率。

题目描述

- 一年中有 2^n 天，有 k 个人，求至少两个人同一天生日的概率。
- $1 \le n \le 10^{18}$，$1 \le k \le 10^{18}$

题目描述

- 一年中有 2^n 天，有 k 个人，求至少两个人同一天生日的概率。
- $1 \le n \le 10^{18}$，$1 \le k \le 10^{18}$
- 输出最简分数，分子分母在化为最简之后膜 $10^6 + 3$ 输出。

提示

- 可以从反面考虑？！

题解

- 首先我们考虑至少两个人同一天生日的概率，事实上就是 $1-$任何一个人生日都不同的概率

题解

- 首先我们考虑至少两个人同一天生日的概率，事实上就是 $1-$任何一个人生日都不同的概率
- 那任何一个人生日都不同的概率呢？

题解

- 首先我们考虑至少两个人同一天生日的概率，事实上就是 $1-$任何一个人生日都不同的概率
- 那任何一个人生日都不同的概率呢？
- 我们还是来钦点一位同学来讲一下（不限数据范围）。

题解

- 考虑任何一个人生日都不同的概率，事实上就是从 2^n 天中有顺序的选出 k 天，就是 $P_{2^n}^{k}$

题解

- 考虑任何一个人生日都不同的概率，事实上就是从 2^n 天中有顺序的选出 k 天，就是 $P_{2^n}^k$
- 分母就是从 2^n 天中任意选取 k 天的方案数，就是2^{nk}

题解

- 考虑任何一个人生日都不同的概率，事实上就是从 2^n 天中有顺序的选出 k 天，就是 $P_{2^n}^k$
- 分母就是从 2^n 天中任意选取 k 天的方案数，就是2^{nk}
- 总的答案就是

题解

- 考虑任何一个人生日都不同的概率，事实上就是从 2^n 天中有顺序的选出 k 天，就是 $P_{2^n}^k$
- 分母就是从 2^n 天中任意选取 k 天的方案数，就是2^{nk}
- 总的答案就是
- $$1 - \frac{\prod_{i=1}^{k-1}(2^n - i)}{2^{n(k-1)}}$$

题解

- 考虑任何一个人生日都不同的概率，事实上就是从 2^n 天中有顺序的选出 k 天，就是 $P_{2^n}^k$
- 分母就是从 2^n 天中任意选取 k 天的方案数，就是2^{nk}
- 总的答案就是
- $$1-\frac{\prod_{i=1}^{k-1}(2^n-i)}{2^{n(k-1)}}$$
- 要怎样化简这个式子呢？

题解

- 首先我们发现分母是 2 的次幂

题解

- 首先我们发现分母是 2 的次幂
- 我们只需从分子中提取 2 的次幂就可以了

题解

- 首先我们发现分母是 2 的次幂
- 我们只需从分子中提取 2 的次幂就可以了
- 发现一个规律：i 和 $2^n - i$ 含有 2 的因数相同，容易证明

题解

- 首先我们发现分母是 2 的次幂
- 我们只需从分子中提取 2 的次幂就可以了
- 发现一个规律：i 和 $2^n - i$ 含有 2 的因数相同，容易证明
- 现在只要求出 $[1, k-1]$ 内有多少个 2 的因子就OK了。

题解

- 那如何求出 $(k-1)!$ 内因子 2 的个数呢？

题解

- 那如何求出 $(k-1)!$ 内因子 2 的个数呢？
- 考虑在 $[1, k-1]$ 内，有$\left\lfloor \frac{k-1}{2^1} \right\rfloor$个数能被$2^1$整除，有$\left\lfloor \frac{k-1}{2^2} \right\rfloor$个数能被$2^2$整除，有.......

题解

- 那如何求出 $(k-1)!$ 内因子 2 的个数呢？
- 考虑在 $[1, k-1]$ 内，有$\left\lfloor \frac{k-1}{2^1} \right\rfloor$个数能被$2^1$整除，有$\left\lfloor \frac{k-1}{2^2} \right\rfloor$个数能被$2^2$整除，有……
- 那么总的式子就是

题解

- 那如何求出 $(k-1)!$ 内因子 2 的个数呢？
- 考虑在 $[1, k-1]$ 内，有$\left\lfloor \frac{k-1}{2^1} \right\rfloor$个数能被$2^1$整除，有$\left\lfloor \frac{k-1}{2^2} \right\rfloor$个数能被$2^2$整除，有.......
- 那么总的式子就是
- $$\sum_{i=1}^{\infty} \left\lfloor \frac{k-1}{2^i} \right\rfloor$$

题解

- 那如何求出 $(k-1)!$ 内因子 2 的个数呢？
- 考虑在 $[1, k-1]$ 内，有 $\left\lfloor \frac{k-1}{2^1} \right\rfloor$ 个数能被 2^1 整除，有 $\left\lfloor \frac{k-1}{2^2} \right\rfloor$ 个数能被 2^2 整除，有.......
- 那么总的式子就是
- $$\sum_{i=1}^{\infty} \left\lfloor \frac{k-1}{2^i} \right\rfloor$$
- 这一步可以 $O(log_2 k)$ 完成了

题解

- 最后约分时，分母直接减幂次，分子用费马小定理或扩欧暴力求逆元，这样就完成了「膜」意义下的约分。

题解

- 最后约分时，分母直接减幂次，分子用费马小定理或扩欧暴力求逆元，这样就完成了「膜」意义下的约分。
- 复杂度 $O(log_2 n + log_2 k)$

Thanks For Your Listening!!!