

二分&倍增浅谈

by 冯青同



知识点一览

- 1、二分答案类型题目的解决通法
- 2、利用二分扩展线段树的功能
- 3、整体二分
- 4、倍增



例0 猜数游戏

- 老师我啊，心里想了一个数字，这个数字在 1 到 1024 之间，请你们想一想，应该怎样在 10 次之内猜出来老师究竟想的是哪个数？
- 这是个互动环节，虽然我觉得没啥互动的必要（因为你们已经学过二分子
- 不过呢，我们不妨来看看，在猜数游戏的过程中，我们实际上都做了什么：
- 1、确定上下界，二分答案。这个就是我们猜答案的方法嘛，按照一半一半的划分猜。
- 2、检查我们猜的答案的正确性。这也就是问老师：老师老师，我猜的是大了、小了，还是正好猜中？
- 3、判断我们下一步应该往哪个方向猜。
- 二分答案的题无非就这三个步骤，接下来，我们用例题来细致讲解：

例1 汽车拉力比赛

题目描述

 复制Markdown  展开

博艾市将要举行一场汽车拉力比赛。

赛场凹凸不平，所以被描述为 $M \times N$ 的网格来表示海拔高度($1 \leq M, N \leq 500$)，每个单元格的海拔范围在0到 10^9 之间。

其中一些单元格被定义为路标。组织者希望给整个路线指定一个难度系数 D ，这样参赛选手从任一路标到达别的路标所经过的路径上相邻单元格的海拔高度差不会大于 D 。也就是说这个难度系数 D 指的是保证所有路标相互可达的最小值。任一单元格和其东西南北四个方向上的单元格都是相邻的。

例1 汽车拉力比赛

- 首先，识别题目：这道题目的本质是，使相邻两个各自的海拔最大值最小。这就是二分答案类型题目的典型特征。
- 接着，确定一下答案的上界和下界：设 $\min d$ 为所有 D_i 中的最小值， $\max d$ 为所有 D_i 中的最大值，则我们二分答案的区间即为 $[\min d, \max d]$
- 然后，考虑如何判定答案是否可行。其实也就是判定所有的路标能否相互到达。我们考虑这样建模：把每个格子看作一个点，向与之直接相邻的四个格子连边，权值为海拔之差。设要判定的答案为 mid ，我们采用BFS的方法判定。以任意一个路标为起点，只有权值小于等于 mid 的边可以扩展节点，看能不能到达所有的路标即可。
- 注：这里使用并查集也可以哦。
- 最后，显然答案越大越好，所以如果 mid 可行，那就把令 $r=\text{mid}-1$ ；否则，令 $l=\text{mid}+1$ 。

例2 包裹快递

题目描述

 复制Markdown  展开

小 K 成功地破解了密文。但是乘车到 X 国的时候，发现钱包被偷了，于是无奈之下只好作快递员来攒足路费去 Orz 教主.....

一个快递公司要将 n 个包裹分别送到 n 个地方，并分配给邮递员小 K 一个事先设定好的路线，小 K 需要开车按照路线给的地点顺序相继送达，且不能遗漏一个地点。小 K 得到每个地方可以签收的时间段，并且也知道路线中一个地方到下一个地方的距离。若到达某一个地方的时间早于可以签收的时间段，则必须在这个地方停留至可以签收，但不能晚于签收的时间段，可以认为签收的过程是瞬间完成的。

为了节省燃料，小 K 希望在全部送达的情况下，车的最大速度越小越好，就找到了你给他设计一种方案，并求出车的最大速度最小是多少。

例2 包裹快递

- “最大速度最小”，很明显符合二分答案题目的特征。
- 答案的上界和下界是多少？下界设为0，上界设为一个足够大的数即可。
- 但是，该怎样二分小数答案？直接令 l, r 为 long double 类型，把终止条件设为 $r - l \leq 1e-6$ 是一种可行的想法，另一种想法则是：直接枚举乘了 10^6 以后的数，在 check 的时候除以 1000000.0
- 检验答案是否可行怎么办呢？我们发现，样例给出的方法完全是不必要的：贪心地来思考，我们每一段路都用最大速度跑一定是更优的。提前到了怎么办？等到规定的时间即可。这是个简单的模拟。
- 假设 mid 可行，那么大于 mid 的一定可行，令 $r = mid - 1$ 即可；如若 mid 不可行，令 $l = mid + 1$ 即可。

例3 “聪明”的质检员

题目描述

[M+](#) [复制Markdown](#) [\[:\]](#) [展开](#)

小T 是一名质量监督员，最近负责检验一批矿产的质量。这批矿产共有 n 个矿石，从 1 到 n 逐一编号，每个矿石都有自己的重量 w_i 以及价值 v_i 。检验矿产的流程是：

- 1、给定 m 个区间 $[l_i, r_i]$ ；
- 2、选出一个参数 W ；
- 3、对于一个区间 $[l_i, r_i]$ ，计算矿石在这个区间上的检验值 y_i ：

$$y_i = \sum_{j=l_i}^{r_i} [w_j \geq W] \times \sum_{j=l_i}^{r_i} [w_j \geq W] v_j$$

其中 j 为矿石编号。

这批矿产的检验结果 y 为各个区间的检验值之和。即： $\sum_{i=1}^m y_i$

若这批矿产的检验结果与所给标准值 s 相差太多，就需要再去检验另一批矿产。小T 不想费时间去检验另一批矿产，所以他想通过调整参数 W 的值，让检验结果尽可能的靠近标准值 s ，即使得 $|s - y|$ 最小。请你帮忙求出这个最小值。

例3 “聪明”的质检员

- 容易发现， W 越小， y 越大，我们不妨考虑二分出令 y 小于 s 的最小的 W ，然后再把 $W-1$ 、 W 两个数带入式子算入的 $|s-y|$ 取个 \min 即可。
- 还是考虑二分答案位于区间 $[l,r]$ ，每次计算 y ，若 $y \geq s$ ，则 $l = \text{mid} + 1$ ；否则令 $r = \text{mid} - 1$ 。
- 接下来，考虑如何计算 y 即可。
- 容易发现， y 的形式是两个 Σ 相乘的形式，那么我们先花 $O(N)$ 的时间预处理两个前缀和数组，然后再花 $O(M)$ 的时间计算一下 y 即可。
- 时间复杂度 $O((N+M)\log|V|)$ ，其中 $|V|$ 代表值域。

例4 New Year's Problem

题意翻译

题目描述

Vlad 有 n 个朋友，每个朋友需要且仅需要 1 个礼物。有 m 家礼物商店，如果在第 i 个商店中为朋友 j 买礼物，朋友 j 将获得 p_{ij} 的快乐值。

由于时间紧迫，Vlad 最多只会在 $n - 1$ 家不同的商店中买礼物。请你使每位朋友能获得的快乐值中的最小值最大。

例4 New Year's Problem

- 二分最小的快乐值 mid ，那么每个商店只有快乐值大于 mid 的才会被考虑，所以每个商店都可以满足若干个朋友。
- 怎么 check 呢？这玩意儿好像是个 NPC 问题啊！！
- 大家注意到这里的“ $n-1$ ”了么？对着它再想想？
- 那么，由鸽笼原理可知，我们的 check 条件是：
- (1) 所有人都至少有一家商店可以满足他
- (2) 有一家商店可以满足至少两个人

例5 赛道修建问题

题目描述

C 城将要举办一系列的赛车比赛。在比赛前，需要在城内修建 m 条赛道。

C 城一共有 n 个路口，这些路口编号为 $1, 2, \dots, n$ ，有 $n - 1$ 条适合于修建赛道的双向通行的道路，每条道路连接着两个路口。其中，第 i 条道路连接的两个路口编号为 a_i 和 b_i ，该道路的长度为 l_i 。借助这 $n - 1$ 条道路，从任何一个路口出发都能到达其他所有的路口。

一条赛道是一组互不相同的道路 e_1, e_2, \dots, e_k ，满足可以从某个路口出发，依次经过道路 e_1, e_2, \dots, e_k （每条道路经过一次，不允许调头）到达另一个路口。一条赛道的长度等于经过的各道路的长度之和。为保证安全，要求每条道路至多被一条赛道经过。

目前赛道修建的方案尚未确定。你的任务是设计一种赛道修建的方案，使得修建的 m 条赛道中长度最小的赛道长度最大（即 m 条赛道中最短赛道的长度尽可能大）

例5 赛道修建

- 首先，最小长度越小，越有可能通过 check，所以二分答案的部分这里就不多说了。
- 考虑如何进行 check。
- 容易发现这是一颗树，那么考虑维护树上路径的方法。每个树上路径一定有一个 lca，那么考虑利用 dfs 枚举 lca 是哪一个节点。
- 考虑到这样一个事实：对于某个节点的任意一个儿子，都一定能且只能传上来一条待匹配的路径，而且由贪心可知，这条路径的长度如果比 mid 大，那么直接自匹配即可；否则加入 set 中。然后就是类似上午讲的乘船问题，从小到大枚举每一条路径，设其权值为 w_i ，在 set 中查询最小的满足 $W + w_i \geq \text{mid}$ 的路径，将它们直接匹配即可。
- 复杂度为 $O(N(\log N)^2)$

例6 排序

题目描述

[M↓ 复制Markdown](#) [展开](#)

在 2016 年，佳媛姐姐喜欢上了数字序列。因而她经常研究关于序列的一些奇奇怪怪的问题，现在她在研究一个难题，需要你来帮助她。

这个难题是这样子的：给出一个 1 到 n 的排列，现在对这个排列序列进行 m 次局部排序，排序分为两种：

- `0 l r` 表示将区间 $[l, r]$ 的数字升序排序
- `1 l r` 表示将区间 $[l, r]$ 的数字降序排序

注意，这里是对**下标**在区间 $[l, r]$ 内的数排序。
最后询问第 q 位置上的数字。

例6 排序

- 操作只有两种：对 $[l,r]$ 内的数进行升序 or 降序排序。直接依据题意进行排序的复杂度少说也是 $O(MN\log N)$ ，只能获得30pts（
- 这道题和二分有什么关系？先思考这样一个问题：假如序列中只有 0 和 1，你会做吗？
- 显然可以用线段树做，复杂度为 $O(M\log N)$ ，对吧？那我们有什么办法把原问题转化为 01 序列的排序问题吗？
- 二分答案！我们把大于等于 mid 的数置为 1，小于 mid 的数置为 0，最后看 q 位置上的数是 0 还是 1 即可。如果是 0，说明 q 位置上的数小于 mid ，令 $l=mid+1$ ；如果是 1，说明 q 位置上的数大于等于 mid ，更新答案，令 $r=mid-1$ 。
- 复杂度为 $O(M\log V|\log N)$

例7 普通平衡树

题目描述

 复制Markdown  展开

您需要写一种数据结构（可参考题目标题），来维护一些数，其中需要提供以下操作：

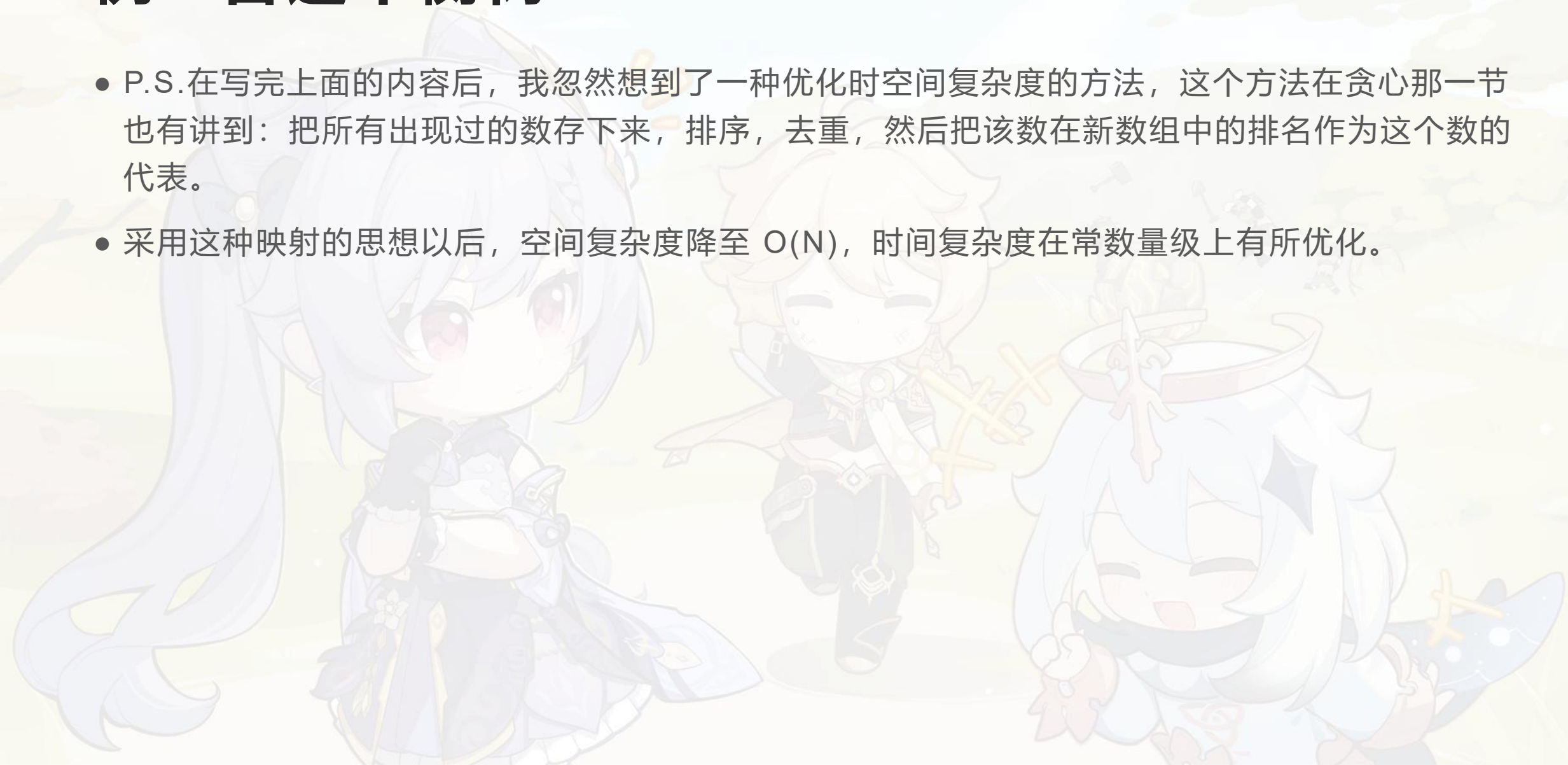
1. 插入 x 数
2. 删除 x 数(若有多多个相同的数，应只删除一个)
3. 查询 x 数的排名(排名定义为比当前数小的数的个数 $+1$)
4. 查询排名为 x 的数
5. 求 x 的前驱(前驱定义为小于 x ，且最大的数)
6. 求 x 的后继(后继定义为大于 x ，且最小的数)

例7 普通平衡树

- 考虑建一颗值域线段树：令线段树的区间代表的含义不再是序列，而是数值。需要维护的是位于区间 $[l,r]$ 的数的总数 sum 。
- 这样的话，修改和查询排名操作都好做，分别是单点修改和区间求和。我们考虑用二分的思想解决求第 k 大数。
- 假设我们现在在 u 节点找第 k 大数，它的左子节点为 ls ，右子节点为 rs 。假如 $sum[ls] \geq k$ ，那么我们就递归去左子节点找第 k 大数；否则，去右子节点找第 $k - sum[ls]$ 大的数。递归终止条件为，找到了一个叶子节点。
- 查前驱就是 $qkth(qrk(x)-1)$ ，查后继就是 $qkth(qrk(x+1))$
- 至此，我们用线段树实现了平衡树的大部分功能。
- 有个小小的问题：这样写的时间复杂度是合格的，但空间复杂度比平衡树多了个 \log ，可能会 MLE。但是胜在好写，真的非常好写。

例7 普通平衡树

- P.S.在写完上面的内容后，我忽然想到了一种优化时空复杂度的方法，这个方法在贪心那一节也有讲到：把所有出现过的数存下来，排序，去重，然后把该数在新数组中的排名作为这个数的代表。
- 采用这种映射的思想以后，空间复杂度降至 $O(N)$ ，时间复杂度在常数量级上有所优化。



例8 货车运输

题目描述

[M↓](#) 复制Markdown [\[\]](#) 展开

A 国有 n 座城市，编号从 1 到 n ，城市之间有 m 条双向道路。每一条道路对车辆都有重量限制，简称限重。

现在有 q 辆货车在运输货物，司机们想知道每辆车在不超过车辆限重的情况下，最多能运多重的货物。

例8 货车运输

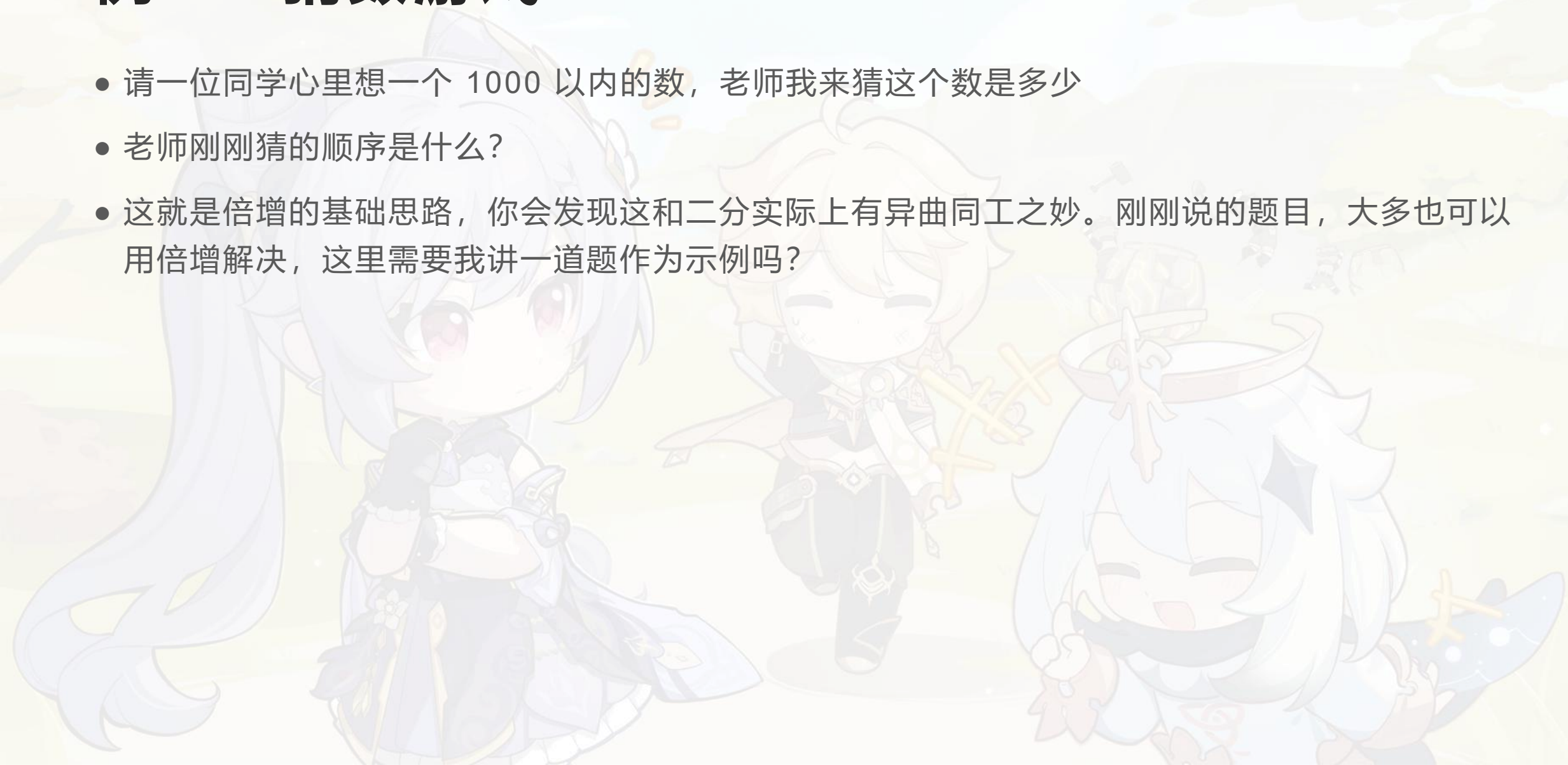
- 这道题是有建一颗最大生成树然后 树剖/倍增 的思路的。这里先简单介绍一下倍增叭。
- 首先，为啥要建最大生成树（其实图可能不连通，最后会建出一个最大生成森林）。因为我们只关心路径上的最小权值，所以如果走了不在最大生成树上的边，答案一定会变小。
- 然后，我相信大家肯定学过倍增 lca，考虑在维护 $fa[i][j]$ 的同时，维护一个 $w[i][j]$ ，表示 i 节点向上跳 2^j 步，经过的边的最小权值，那么 $w[i][j] = \min(w[fa[i][j-1]][j-1], w[i][j-1])$ ，在查询 lca 的过程中顺便维护一下就能求出路径上经过的边的最小权值了。

例8 货车运输

- 接下来考虑“整体二分”的思想。那就是，把所有的询问在同一次二分中完成。
- 首先，对于每一个询问分别二分，复杂度是 $O(qn\log n)$ ，这显然是无法接受的。
- 我们开数组记录下每一个询问的编号、二分的左端点、二分的右端点。考虑到二分最多产生 $\log N$ 层，对于每一层，我们考虑如何减少计算量。来看一下我们的条件：利用并查集维护节点的联通关系；并查集只支持加边；假如权值小的边被加入了，那么权值大的边一定也应该被加入。那么，我们把所有边按照权值排序，这一层的所有询问按照 mid 从大到小排序，按照顺序扫一遍询问并加边，那么我们只需要进行一次并查集的重置操作和 m 次加边操作，复杂度降至 $O((n+m+q)\log N)$ 级别。
- 同时注意到，如果把二分的左、右端点均相同的询问视作一组的话，那么不同组之间的 mid 值的相对大小是不会改变的，而同一组内的 mid 值的变化也很有规律，因此不需要每一层都排序。
- 可能比较抽象，对着代码讲吧。

例8.5 猜数游戏2.0

- 请一位同学心里想一个 1000 以内的数，老师我来猜这个数是多少
- 老师刚刚猜的顺序是什么？
- 这就是倍增的基础思路，你会发现这和二分实际上有异曲同工之妙。刚刚说的题目，大多也可以用倍增解决，这里需要我讲一道题作为示例吗？



例9 跑路

(卡其脱离太)

题目描述

 复制Markdown  展开

小 A 的工作不仅繁琐，更有苛刻的规定，要求小 A 每天早上在 6 : 00 之前到达公司，否则这个月工资清零。可是小 A 偏偏又有赖床的坏毛病。于是为了保住自己的工资，小 A 买了一个空间跑路器，每秒钟可以跑 2^k 千米 (k 是任意自然数)。当然，这个机器是用 `longint` 存的，所以总跑路长度不能超过 `maxlongint` 千米。小 A 的家到公司的路可以看做一个有向图，小 A 家为点 1，公司为点 n ，每条边长度均为一千米。小 A 想每天能醒地尽量晚，所以让你帮他算算，他最少需要几秒才能到公司。数据保证 1 到 n 至少有一条路径。

输入格式

第一行两个整数 n, m ，表示点的个数和边的个数。

接下来 m 行每行两个数字 u, v ，表示一条 u 到 v 的边。

例9 跑路

- n 很小，直接用邻接矩阵存边即可。设 $g[u][v]$ 表示从 u 到 v 是否有长度为 2^k 的路径（注意，这里的 k 为任意值）， $f[u][v][k]$ 表示从 u 到 v 是否有长度为 2^k 的路径。因为保证了最优路径的长度不会超过 `maxlongint`，所以 k 开到 70 就完全足够。
- 转移方程为 $f[u][v][k] = (f[u][v][k] \mid (f[u][l][j-1] \& f[l][v][k-1]))$
- 接下来跑个最短路就行。
- 有同学可能会有疑惑：这样为啥是对的咧？
- 我们这么想：如果没有加速器，那就是若干权值为 1 的边然后跑最短路对吧？但是有加速器，所以一些长度为 2^k 的路径可以只花费 1 的时间就跑到，我们前面的预处理就相当于在所有的能用这种方式跑的一对点之间加了一条权值为 1 的边。

例10 国旗计划

题目描述

 复制Markdown  展开

A 国正在开展一项伟大的计划 —— 国旗计划。这项计划的内容是边防战士手举国旗环绕边境线奔袭一圈。这项计划需要多名边防战士以接力的形式共同完成，为此，国土安全局已经挑选了 N 名优秀的边防战士作为这项计划的候选人。

A 国幅员辽阔，边境线上设有 M 个边防站，顺时针编号 1 至 M 。每名边防战士常驻两个边防站，并且善于在这两个边防站之间长途奔袭，我们称这两个边防站之间的路程是这个边防战士的奔袭区间。 N 名边防战士都是精心挑选的，身体素质极佳，所以每名边防战士的奔袭区间都不会被其他边防战士的奔袭区间所包含。

现在，国土安全局局长希望知道，至少需要多少名边防战士，才能使得他们的奔袭区间覆盖全部的边境线，从而顺利地完国旗计划。不仅如此，安全局局长还希望知道更详细的信息：对于每一名边防战士，在他必须参加国旗计划的前提下，至少需要多少名边防战士才能覆盖全部边境线，从而顺利地完国旗计划。

例10 国旗计划

- 首先给大家科普一下处理环的方式：断环成链。
- 因为所有的战士的奔袭区间都不会被另一名战士所包含，所以如果我们按照左端点递增排序的话，所有战士的右端点也是递增的（注意，如果右端点比左端点小，那么，我们把右端点的值 $+m$ ）
- 暴力的想法肯定是，一个个战士接力，看什么时候最后一个战士的右端点的值第一次比第一个战士的左端点的值大 m ，这时经过的战士数就是我们要求的答案。
- 这样做的时间复杂度是 $O(N^2)$ 的，使用倍增加速这个过程即可。时间复杂度为 $O(N \log N)$ 。

例11 火枪打怪

题目描述

[M↓ 复制Markdown](#) [\[\] 展开](#)

LXL 进入到了一片丛林,结果他发现共有 n 只怪物排成一排站在他面前。LXL 有一杆火枪能对付这些怪物。他知道从左至右数第 i 只怪物的血量是 m_i 。现在 LXL 可以将一些子弹射向某个怪物。LXL 可以控制他所发射的子弹数量及子弹的威力值。当某个子弹射到第 i 个怪物, 如果这个子弹的威力值为 p , 除了这个怪物会掉 p 点血以外, 它左边的第 j 个怪物 ($j < i$), 也会遭到 $\max(0, p - (i - j)^2)$ 的溅射伤害 (好神奇的子弹)。当某只怪物的血量小于 0 时, 它就死了, 但它的尸体还在, 即怪物的位置永远不会改变。LXL 希望只用 k 发子弹, 请你求出一个最小的正整数 p , 使 LXL 用 k 发子弹且每发子弹的威力值为 p 就可以消灭所有怪物。

例11 火枪打怪

- 二分不说，这里只讲如何check。
- 首先，溅射只会往左边溅射，所以考虑从右往左扫，这样就没有后效性。
- 然后，只需要考虑如何维护 $(i-j)^2$ 即可。
- 先求出 mid 的平方根 x ，这里我们来推一下式子：
- 所以只需要维护三个后缀和就行
- $\text{cnt}(l)$ 表示第 l 个怪被打了多少枪

$$\begin{aligned} S &= \sum_{l=j+1}^{l \leq j+x} \text{cnt}_l \times (L-j)^2 \\ &= \sum \text{cnt}_l \cdot L^2 - 2j \cdot \sum \text{cnt}_l \cdot L \\ &\quad + j^2 \cdot \sum \text{cnt}_l. \end{aligned}$$

例12 天路

题目描述

 复制Markdown  展开

“那是一条神奇的天路诶 ~~~，把第一个神犇送上天堂 ~~~”，XDM 先生唱着这首“亲切”的歌曲，一道猥琐题目的灵感在脑中出现了。

和 C_SUNSHINE 大神商量后，这道猥琐的题目终于出现在本次试题上了，旨在难到一帮大脑不够灵活的 OIer 们（JOHNKRAM 真的不是说你.....）。

言归正传，小 X 的梦中，他在西藏开了一家大型旅游公司，现在，他要为西藏的各个景点设计一组铁路线。但是，小 X 发现，来旅游的游客都很挑剔，他们乘火车在各个景点间游览，景点的趣味当然是不用说啦，关键是路上。试想，若是乘火车一圈转悠，却发现回到了游玩过的某个景点，花了一大堆钱却在路上看不到好的风景，那是有多么的恼火啊。

所以，小 X 为所有的路径定义了两个值， V_i 和 P_i ，分别表示火车线路的风景趣味度和乘坐一次的价格。现在小 X 想知道，乘客从任意一个景点开始坐火车走过的一条回路上所有的 V 之和与 P 之和的比值的最大值。以便为顾客们推荐一条环绕旅游路线（路线不一定包含所有的景点，但是不可以存在重复的火车路线）。

于是，小 X 梦醒之后找到了你.....

例12 天路

- 这是一道很经典的题目，我们采用二分答案的方法来解决。
- 我们二分一个答案 mid ，然后要 check 答案 mid 是否可行，怎样 check 呢？
- 如果存在一个环，满足 $\sum v / \sum p \geq ans$ ，那么移项一下就有 $\sum v \geq ans * \sum p$ ，就有 $\sum v \geq \sum ans * p$ ，也即存在一个环满足 $\sum ans * p - v \leq 0$ ，也就是把 $ans * p - v$ 作为新的边权，要判断图中是否存在负环。
- 大家会找负环么？
- spfa 的 bfs 版本和 dfs 版本的介绍。

例13 Telephone Lines S

多年以后，笨笨长大了，成为了电话线布置师。由于地震使得某市的电话线全部损坏，笨笨是负责接到震中市的负责人。该市周围分布着 $1 \leq N \leq 1000$ 根据 $1 \cdots N$ 顺序编号的废弃的电话线杆，任意两根线杆之间没有电话线连接，一共有 $1 \leq p \leq 10000$ 对电话杆可以拉电话线。其他的由于地震使得无法连接。

第 i 对电线杆的两个端点分别是 a_i, b_i ，它们的距离为 $1 \leq l_i \leq 1000000$ 。数据中每对 (a_i, b_i) 只出现一次。编号为 1 的电话杆已经接入了全国的电话网络，整个市的电话线全都连到了编号 N 的电话线杆上。也就是说，笨笨的任务仅仅是找一条将 1 号和 N 号电线杆连起来的路径，其余的电话杆并不一定要连入电话网络。

电信公司决定支援灾区免费为此市连接 k 对由笨笨指定的电话线杆，对于此外的那些电话线，需要为它们付费，总费用决定于其中最长的电话线的长度（每根电话线仅连接一对电话线杆）。如果需要连接的电话线杆不超过 k 对，那么支出为 0。

请你计算一下，将电话线引导震中市最少需要在电话线上花多少钱？

例13 Telephone Lines S

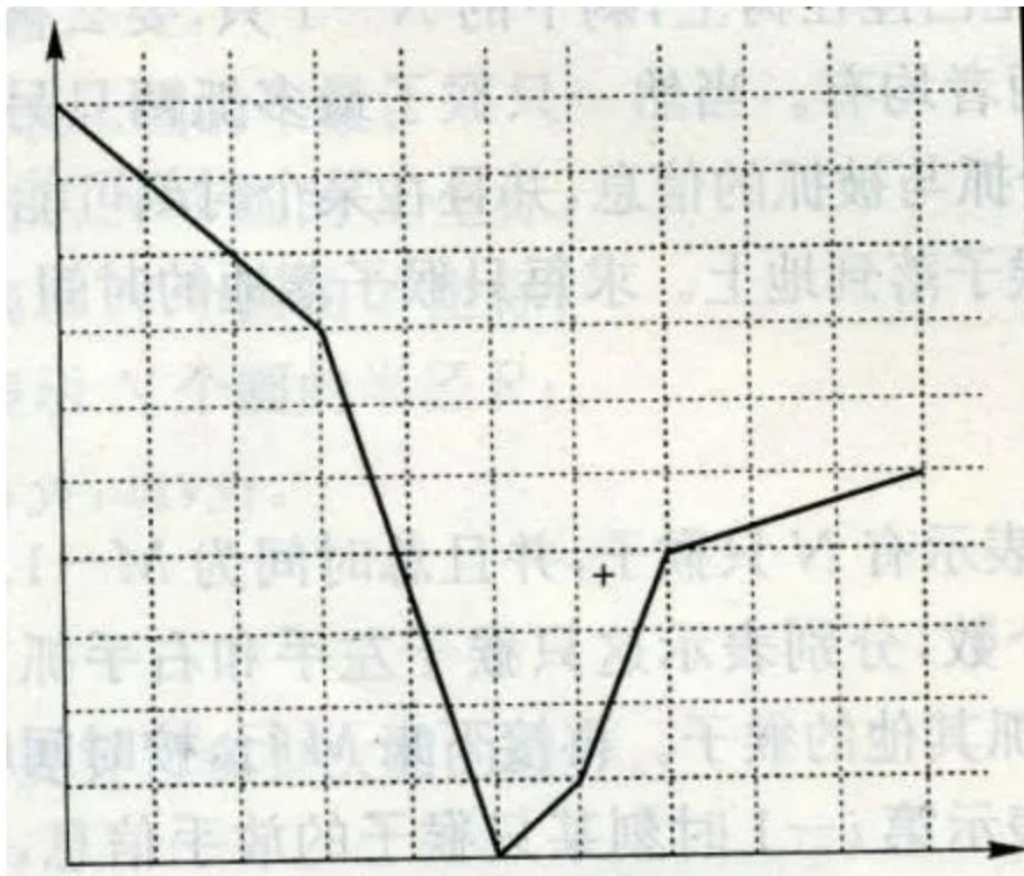
- 这里介绍多种做法：
- 1、二分答案法。二分最大边权为 mid ，把边权大于 mid 的设为代价为 1 的边，边权小于 mid 的设为代价为 0 的边。然后只需求从 1 到 n 的最短路，看看最短路的代价是否比 k 小即可。
- 求解最短路的方法：
- (1) spfa
- (2) 双端队列BFS（放在搜索那块讲）
- 2、分层图最短路法。

例14 山

题目描述

给出一座山，如图。

[M+](#) 复制Markdown [展开](#)



现在要在山上的某个部位装一盏灯，使得这座山的任何一个部位都能够被看到。

给出最小的 y 坐标，如图的 $+$ 号处就是 y 坐标最小的安装灯的地方。

例14 山

- 这其实是个平面几何题？如果一条线能够被照亮，那么安装灯的位置一定在这条线段以上。
- 如何判断一个点在线段以上还是以下？设点为 (x,y) ，把 x 带入直线方程里算一算 y' ，如果 $y > y'$ ，那么点就在线段以上。
- 如何 check 答案？考虑对于二分的高度 mid ，每条直线都只有一部分直线上的点满足高度小于 mid 。这些点可以构成区间的形式，只需要判断区间是否有交集即可。

