

堆解丑数问题



堆方法如何确保生成的丑数是按顺序的？这个怎么证明我还是不理解

推理关于堆生成丑数的顺序，持续 12 秒

我们可以用数学归纳法来证明，用堆（最小堆）生成丑数的过程中，每次取出的数都是当前所有可能丑数中的最小值，从而保证生成的序列是递增的。下面是详细证明过程：

证明思路：数学归纳法

1. 定义与初始化

- 我们定义丑数序列为： a_1, a_2, a_3, \dots ，其中 $a_1 = 1$ 。
- 算法一开始将 1 放入最小堆中。此时堆只包含一个元素 1，也就是序列的第一个丑数。

2. 归纳假设

假设经过若干次操作，我们已经从堆中依次取出了前 k 个丑数，且它们是严格递增的：

$$a_1 < a_2 < \dots < a_k$$

并且这 k 个数正是丑数序列中最小的 k 个数。

3. 归纳步骤

接下来我们证明第 $k + 1$ 个丑数也会是序列中剩下丑数中最小的。

• 候选数的生成

当我们从堆中取出当前最小的数 a_i （在前 k 个中，已经被取出的数），我们会将 $a_i \times 2$ 、 $a_i \times 3$ 和 $a_i \times 5$ 作为新候选丑数加入堆中（前提是这些数之前没有加入过）。

• 堆的性质保证最小候选被提取

由于堆始终维护当前所有候选数的最小值，下一次从堆中取出的数一定是所有候选数中的最小值。

注意，候选数都是由已确定的丑数（即 a_1 到 a_k ）乘以 2、3 或 5 得到的，因此所有候选数都必然大于或等于 a_k 。

• 证明没有遗漏更小的丑数

假设存在某个丑数 x 且 $a_k < x <$ （下一个从堆中取出的数），那么 x 必定可以写成某个丑数 a_i （其中 $1 \leq i \leq k$ ）乘以 2、3 或 5 得到。这说明 x 应该已经出现在候选数中，但由于堆总是取最小的候选数，这与假设矛盾。因此不存在这样的 x 。

• 结论

因此，堆中取出的下一个数，即第 $k + 1$ 个丑数，必然是所有还未生成的丑数中最小的，从而保证了整个

序列的严格递增性。

总结

1. **初始化**：从 1 开始，作为第一个丑数。
2. **候选生成**：每次从堆中取出当前最小的丑数 a_i ，并生成候选数 $a_i \times 2$ 、 $a_i \times 3$ 和 $a_i \times 5$ 。
3. **堆的性质**：由于最小堆总是保证当前所有候选数中最小的数被首先提取，这保证了每次得到的数都是序列中下一个最小的丑数。
4. **归纳证明**：假设前 k 个丑数都是正确且递增的，那么第 $k + 1$ 个数必然也是所有未取候选数中的最小值，从而继续保持递增顺序。

这种方式相当于按“层次”生成丑数，每次只扩展当前已经找到的丑数的倍数，从而确保不会跳过任何比当前候选数更小的丑数。这就是为什么用堆的方法能**保证生成的丑数序列始终是有序的**。

希望这个详细的归纳证明能帮助你更好地理解堆方法的顺序保证原理！如果还有疑问，欢迎继续讨论。