

# 哈希的应用

张超

南京外国语学校

2022 年 12 月 5 日



① 例题 1

② 例题 2

③ 例题 3

④ 例题 4

⑤ 例题 5

⑥ 例题 6

⑦ 例题 7

⑧ 例题 8

⑨ 例题 9



# 1 例题 1

## 2 例题 2

## 3 例题 3

## 4 例题 4

## 5 例题 5

## 6 例题 6

## 7 例题 7

## 8 例题 8

## 9 例题 9



- 给定 5 个长度为  $n$  的数组，判断能否从每个数组中选出一个数，使得和为 0，数值访问  $[-10^{15}, 10^{15}]$ ,  $n \leq 200$ 。  
(来源: HDU4334)



- 直接循环枚举，复杂度  $O(n^5)$ ，超时。



- 直接循环枚举，复杂度  $O(n^5)$ ，超时。
- 如果数值范围比较小，可以怎么做？



- 直接循环枚举，复杂度  $O(n^5)$ ，超时。
- 如果数值范围比较小，可以怎么做？
- 解法 1：考虑将 5 个数组划分成两部分，前 2 个数组和后 3 个数组。  
把前 2 个数组中两个数的和  $A[i] + B[j]$  存入到容器  $S$  中。  
后三个数组  $C, D, E$  中，每个数组选择一个数。判断  
 $-(C[x] + D[y] + E[z])$  是否在容器中出现过。



- 直接循环枚举，复杂度  $O(n^5)$ ，超时。
- 如果数值范围比较小，可以怎么做？
- 解法 1：考虑将 5 个数组划分成两部分，前 2 个数组和后 3 个数组。  
把前 2 个数组中两个数的和  $A[i] + B[j]$  存入到容器  $S$  中。  
后三个数组  $C, D, E$  中，每个数组选择一个数。判断  
 $-(C[x] + D[y] + E[z])$  是否在容器中出现过。
- 容器  $S$ ，可以通过外拉链解决 hash 冲突来实现。





- 直接循环枚举，复杂度  $O(n^5)$ ，超时。
- 如果数值范围比较小，可以怎么做？
- 解法 1：考虑将 5 个数组划分成两部分，前 2 个数组和后 3 个数组。  
把前 2 个数组中两个数的和  $A[i] + B[j]$  存入到容器  $S$  中。  
后三个数组  $C, D, E$  中，每个数组选择一个数。判断  
 $-(C[x] + D[y] + E[z])$  是否在容器中出现过。
- 容器  $S$ ，可以通过外拉链解决 hash 冲突来实现。
- 总复杂度  $O(n^3)$



- 解法 2: 把 5 个集合划分成 3 个部分,  
 $A[i], W = B[j] + C[k], V = -(D[x] + E[y])$



- 解法 2: 把 5 个集合划分成 3 个部分,  
 $A[i], W = B[j] + C[k], V = -(D[x] + E[y])$
- 将三个数组都排序, 枚举  $A[i]$ , 扫描  $W$  和  $V$  数组,  $A[i] + W[j]$  是单调递增的,  $V[k]$  也是单调递增的, 可以进行归并



- 解法 2: 把 5 个集合划分成 3 个部分,  
 $A[i], W = B[j] + C[k], V = -(D[x] + E[y])$
- 将三个数组都排序, 枚举  $A[i]$ , 扫描  $W$  和  $V$  数组,  $A[i] + W[j]$  是单调递增的,  $V[k]$  也是单调递增的, 可以进行归并
- 对于每个  $i$ , 从头开始滑动  $j$  和  $k$ , 归并  $A[i] + W[j]$  和  $V[k]$



- 解法 2: 把 5 个集合划分成 3 个部分,  
 $A[i], W = B[j] + C[k], V = -(D[x] + E[y])$
- 将三个数组都排序, 枚举  $A[i]$ , 扫描  $W$  和  $V$  数组,  $A[i] + W[j]$  是单调递增的,  $V[k]$  也是单调递增的, 可以进行归并
- 对于每个  $i$ , 从头开始滑动  $j$  和  $k$ , 归并  $A[i] + W[j]$  和  $V[k]$
- 复杂度  $O(n^3)$



1 例题 1

2 例题 2

3 例题 3

4 例题 4

5 例题 5

6 例题 6

7 例题 7

8 例题 8

9 例题 9



- 已知多项式方程：

$$a_0 + a_1x + a_2x^2 + \cdots + a_nx^n = 0$$

求这个方程在  $[1, m]$  内的整数解 ( $n$  和  $m$  均为正整数)。

$0 < n \leq 100, |a_i| \leq 10^{10000}, a_n \neq 0, m < 10^6$ 。

(来源：NOIP2014 解方程)



- 如果  $t = 0$  则有  $t \% p = 0$ , 但是如果  $t \% p = 0$ , 则  $t = 0$  不一定成立





- 如果  $t = 0$  则有  $t \% p = 0$ , 但是如果  $t \% p = 0$ , 则  $t = 0$  不一定成立
- 如果  $t \% p_i, 1 \leq i \leq 5$  都是 0,  $t$  就很大概率为 0



- 如果  $t = 0$  则有  $t \% p = 0$ , 但是如果  $t \% p = 0$ , 则  $t = 0$  不一定成立
- 如果  $t \% p_i, 1 \leq i \leq 5$  都是 0,  $t$  就很大概率为 0
- $f(x) \% p = f(x \% p) \% p$



- 如果  $t = 0$  则有  $t \% p = 0$ , 但是如果  $t \% p = 0$ , 则  $t = 0$  不一定成立
- 如果  $t \% p_i, 1 \leq i \leq 5$  都是 0,  $t$  就很大概率为 0
- $f(x) \% p = f(x \% p) \% p$
- 可以先把较小的  $[0, p_i - 1]$ , 这些  $f[i][j]$  预处理出来



- 如果  $t = 0$  则有  $t \% p = 0$ , 但是如果  $t \% p = 0$ , 则  $t = 0$  不一定成立
- 如果  $t \% p_i, 1 \leq i \leq 5$  都是 0,  $t$  就很大概率为 0
- $f(x) \% p = f(x \% p) \% p$
- 可以先把较小的  $[0, p_i - 1]$ , 这些  $f[i][j]$  预处理出来
- 枚举  $x$  是否可行时, 只需要判断  $f[i][x \% p_i]$  是否都为 0 即可



1 例题 1

2 例题 2

3 例题 3

4 例题 4

5 例题 5

6 例题 6

7 例题 7

8 例题 8

9 例题 9



- 给定一个长度为  $n$  的序列，有  $m$  个询问，每次询问区间  $[l, r]$  是否为 1 到  $r - l + 1$  的一个排列。 $n, m \leq 10^6$ 。  
(题目来源: HDU5172)



- 考虑如何快速取出一段区间，且顺序不影响结果。



- 考虑如何快速取出一段区间，且顺序不影响结果。
- 求和太容易出错了。可以使用异或





- 考虑如何快速取出一段区间，且顺序不影响结果。
- 求和太容易出错了。可以使用异或
- 给每个数对应一个随机值。求出  $1 \sim i$  每个前缀的哈希值  $t[i]$



- 考虑如何快速取出一段区间，且顺序不影响结果。
- 求和太容易出错了。可以使用异或
- 给每个数对应一个随机值。求出  $1 \sim i$  每个前缀的哈希值  $t[i]$
- 再得到序列的异或前缀和  $s[i]$ ，则  $[l, r]$  区间的 hash 值，为  $s[r] \oplus s[l-1]$ ，判断与  $t[r-l+1]$  是否相等



- 考虑如何快速取出一段区间，且顺序不影响结果。
- 求和太容易出错了。可以使用异或
- 给每个数对应一个随机值。求出  $1 \sim i$  每个前缀的哈希值  $t[i]$
- 再得到序列的异或前缀和  $s[i]$ ，则  $[l, r]$  区间的 hash 值，为  $s[r] \oplus s[l-1]$ ，判断与  $t[r-l+1]$  是否相等
- 可以多取几组随机值。



1 例题 1

2 例题 2

3 例题 3

4 例题 4

5 例题 5

6 例题 6

7 例题 7

8 例题 8

9 例题 9



- 给定一个有  $n$  个点  $m$  条边的有向图，有  $q$  次操作，每种操作属于以下四种之一：
  - 把一条边删掉
  - 把一个点所有的入边都删掉
  - 把一条边恢复
  - 把一个点的所有入边恢复每次操作完成之后判断每个点的是否都只有一条出边。  
 $n, m, q \leq 5 \times 10^5$   
(题目来源: CSP2022 星战)



- 如果从  $i$  出发的边，给它一个特定的随机值  $A[i]$



- 如果从  $i$  出发的边，给它一个特定的随机值  $A[i]$
- 判断所有点是否都为 1，就是判断所有点的出边和是否为  $P = \sum A[i]$



- 如果从  $i$  出发的边，给它一个特定的随机值  $A[i]$
- 判断所有点是否都为 1，就是判断所有点的出边和是否为  $P = \sum A[i]$
- 只需要维护所有点的出边权值和  $S$ 。





- 如果从  $i$  出发的边，给它一个特定的随机值  $A[i]$
- 判断所有点是否都为 1，就是判断所有点的出边和是否为  $P = \sum A[i]$
- 只需要维护所有点的出边权值和  $S$ 。
- 考虑需要对一个点的入边集合统一处理，记录一个点的初始入边和为  $B[i] = \sum A[x]$ , 有  $x$  到  $i$  的一条边



- 如果从  $i$  出发的边，给它一个特定的随机值  $A[i]$
- 判断所有点是否都为 1，就是判断所有点的出边和是否为  $P = \sum A[i]$
- 只需要维护所有点的出边权值和  $S$ 。
- 考虑需要对一个点的入边集合统一处理，记录一个点的初始入边和为  $B[i] = \sum A[x]$ , 有  $x$  到  $i$  的一条边
- 当前入边和为  $C[i]$   $S = \sum C[i]$  (入边总和与出边总和相同)



- 如果从  $i$  出发的边，给它一个特定的随机值  $A[i]$
- 判断所有点是否都为 1，就是判断所有点的出边和是否为  $P = \sum A[i]$
- 只需要维护所有点的出边权值和  $S$ 。
- 考虑需要对一个点的入边集合统一处理，记录一个点的初始入边和为  $B[i] = \sum A[x]$ , 有  $x$  到  $i$  的一条边
- 当前入边和为  $C[i]$   $S = \sum C[i]$  (入边总和与出边总和相同)
- 对一条边进行操作时，修改  $S$  和  $C[i]$



- 如果从  $i$  出发的边，给它一个特定的随机值  $A[i]$
- 判断所有点是否都为 1，就是判断所有点的出边和是否为  $P = \sum A[i]$
- 只需要维护所有点的出边权值和  $S$ 。
- 考虑需要对一个点的入边集合统一处理，记录一个点的初始入边和为  $B[i] = \sum A[x]$ ，有  $x$  到  $i$  的一条边
- 当前入边和为  $C[i]$   $S = \sum C[i]$  (入边总和与出边总和相同)
- 对一条边进行操作时，修改  $S$  和  $C[i]$
- 对一个点的所有入边删除:  $S - = C[i], C[i] = 0$



- 如果从  $i$  出发的边，给它一个特定的随机值  $A[i]$
- 判断所有点是否都为 1，就是判断所有点的出边和是否为  $P = \sum A[i]$
- 只需要维护所有点的出边权值和  $S$ 。
- 考虑需要对一个点的入边集合统一处理，记录一个点的初始入边和为  $B[i] = \sum A[x]$ , 有  $x$  到  $i$  的一条边
- 当前入边和为  $C[i]$   $S = \sum C[i]$  (入边总和与出边总和相同)
- 对一条边进行操作时，修改  $S$  和  $C[i]$
- 对一个点的所有入边删除:  $S - = C[i], C[i] = 0$
- 对一个点的恢复所有入边时:  $S + = B[i] - C[i], C[i] = B[i]$



1 例题 1

2 例题 2

3 例题 3

4 例题 4

5 例题 5

6 例题 6

7 例题 7

8 例题 8

9 例题 9



- 给定一个长度为  $n$  的字符串  $A$  和长度为  $m$  的字符串  $B$ , 判断  $B$  是否为  $A$  的子串。 $n, m \leq 10^6$   
(来源: 经典问题)



- 直接枚举 A 串的起点，判断与 B 是否相同，复杂度高达  $O(n * m)$ 。





- 直接枚举  $A$  串的起点，判断与  $B$  是否相同，复杂度高达  $O(n * m)$ 。
- 考虑使用 `hash`，把  $B$  串映射成一个值。把  $A$  的每个长度为  $m$  的子串也映射成一个值。判断字符串是否相同，变成判断数值相等。



- 直接枚举  $A$  串的起点，判断与  $B$  是否相同，复杂度高达  $O(n * m)$ 。
- 考虑使用 hash，把  $B$  串映射成一个值。把  $A$  的每个长度为  $m$  的子串也映射成一个值。判断字符串是否相同，变成判断数值相等。
- 如何快速把  $A$  中的每个子串都映射成一个值。



- 直接枚举 A 串的起点，判断与 B 是否相同，复杂度高达  $O(n * m)$ 。
- 考虑使用 hash，把 B 串映射成一个值。把 A 的每个长度为 m 的子串也映射成一个值。判断字符串是否相同，变成判断数值相等。
- 如何快速把 A 中的每个子串都映射成一个值。
- $S_i = S_{i-1} * Base + A_i$



- 直接枚举 A 串的起点，判断与 B 是否相同，复杂度高达  $O(n * m)$ 。
- 考虑使用 hash，把 B 串映射成一个值。把 A 的每个长度为 m 的子串也映射成一个值。判断字符串是否相同，变成判断数值相等。
- 如何快速把 A 中的每个子串都映射成一个值。
- $S_i = S_{i-1} * Base + A_i$
- $Hash(L, R) = S_R - S_L * Base^{R-L+1}$



- 直接枚举 A 串的起点，判断与 B 是否相同，复杂度高达  $O(n * m)$ 。
- 考虑使用 hash，把 B 串映射成一个值。把 A 的每个长度为 m 的子串也映射成一个值。判断字符串是否相同，变成判断数值相等。
- 如何快速把 A 中的每个子串都映射成一个值。
- $S_i = S_{i-1} * Base + A_i$
- $Hash(L, R) = S_R - S_L * Base^{R-L+1}$
- Base 一般取素数，自然溢出容易被卡，可以选择取模  $P = 10^9 + 7$



- 直接枚举 A 串的起点，判断与 B 是否相同，复杂度高达  $O(n * m)$ 。
- 考虑使用 hash，把 B 串映射成一个值。把 A 的每个长度为 m 的子串也映射成一个值。判断字符串是否相同，变成判断数值相等。
- 如何快速把 A 中的每个子串都映射成一个值。
- $S_i = S_{i-1} * Base + A_i$
- $Hash(L, R) = S_R - S_L * Base^{R-L+1}$
- Base 一般取素数，自然溢出容易被卡，可以选择取模  $P = 10^9 + 7$
- 最终的复杂度是  $O(n + m)$



- 扩展：给定两个长度为  $n$  的字符串  $A$  和  $B$ , 求它们的最长公共子串。  
 $n, m \leq 10^5$



- 如果答案是  $k$ ，那么一定能找到长度为  $k-1$  的公共子串。答案满足单调性。





- 如果答案是  $k$ ，那么一定能找到长度为  $k - 1$  的公共子串。答案满足单调性。
- 使用二分，把  $A$  中长度为  $mid$  的子串  $hash$  值，都记录下来（可以用哈希表实现）。再枚举  $B$  中的长度为  $mid$  的子串，判断其  $hash$  值是否出现过。



- 如果答案是  $k$ ，那么一定能找到长度为  $k - 1$  的公共子串。答案满足单调性。
- 使用二分，把  $A$  中长度为  $mid$  的子串  $hash$  值，都记录下来（可以用哈希表实现）。再枚举  $B$  中的长度为  $mid$  的子串，判断其  $hash$  值是否出现过。
- 复杂度为  $O(n \log n)$



1 例题 1

2 例题 2

3 例题 3

4 例题 4

5 例题 5

6 例题 6

7 例题 7

8 例题 8

9 例题 9



- 给定一个长度为  $n$  的字符串  $A$ , 求它的最长回文子串。  $n \leq 10^5$   
(来源: 经典问题)



- 枚举中心点，向两边扩展 (分奇偶讨论)



- 枚举中心点，向两边扩展（分奇偶讨论）
- 扩展过程，可以用二分代替，找到最远相等的位置。



- 枚举中心点，向两边扩展 (分奇偶讨论)
- 扩展过程，可以用二分代替，找到最远相等的位置。
- 以奇数长度的回文串为例，判断  $A[x-mid, x]$  和  $A[x, x+mid]$  翻转后是否相同 (回文)



- 枚举中心点，向两边扩展（分奇偶讨论）
- 扩展过程，可以用二分代替，找到最远相等的位置。
- 以奇数长度的回文串为例，判断  $A[x-mid, x]$  和  $A[x, x+mid]$  翻转后是否相同（回文）
- 在正串上的得到  $HI$ ，在反串上得到  $Hr$





- 枚举中心点，向两边扩展 (分奇偶讨论)
- 扩展过程，可以用二分代替，找到最远相等的位置。
- 以奇数长度的回文串为例，判断  $A[x-mid, x]$  和  $A[x, x+mid]$  翻转后是否相同 (回文)
- 在正串上的得到  $HI$ ，在反串上得到  $Hr$
- 左边的哈希值从正串中获得，为  $HI[x] - HI[x - mid - 1] * Base[mid + 1]$



- 枚举中心点，向两边扩展 (分奇偶讨论)
- 扩展过程，可以用二分代替，找到最远相等的位置。
- 以奇数长度的回文串为例，判断  $A[x-mid, x]$  和  $A[x, x+mid]$  翻转后是否相同 (回文)
- 在正串上的得到  $HI$ ，在反串上得到  $Hr$
- 左边的哈希值从正串中获得，为  $HI[x] - HI[x - mid - 1] * Base[mid + 1]$
- 右边的哈希值从反串中获得，为  $Hr[x] - Hr[x + mid + 1] * Base[mid + 1]$



1 例题 1

2 例题 2

3 例题 3

4 例题 4

5 例题 5

6 例题 6

7 例题 7

8 例题 8

9 例题 9



- 给定一个长度为  $n$  的字符串  $A$ ，每次可以  $A$  的头部或者尾部取出一个字母，放入到新字符串的尾部，求新的字符串最小的字典序。 $n \leq 10^5$   
(来源：洛谷 P2870)



- 每次比较头部  $A[i]$  和尾部  $A[j]$  两个字母的大小，如果不相等，哪个小用哪个字母。



- 每次比较头部  $A[i]$  和尾部  $A[j]$  两个字母的大小，如果不相等，哪个小用哪个字母。
- 如果相等，继续向内比较，找到第一个不同的位置  $x$  和  $y$ ，如果  $A[x] < A[y]$ ，则  $A[i]$  出来，否则  $A[j]$  出来。



- 每次比较头部  $A[i]$  和尾部  $A[j]$  两个字母的大小，如果不相等，哪个小用哪个字母。
- 如果相等，继续向内比较，找到第一个不同的位置  $x$  和  $y$ ，如果  $A[x] < A[y]$ ，则  $A[i]$  出来，否则  $A[j]$  出来。
- 向内找到第一个不相等的位置，就是找最远相等的位置，可以使用二分，与回文串类似，使用正反串上的 **hash** 值比较。



- 每次比较头部  $A[i]$  和尾部  $A[j]$  两个字母的大小，如果不相等，哪个小用哪个字母。
- 如果相等，继续向内比较，找到第一个不同的位置  $x$  和  $y$ ，如果  $A[x] < A[y]$ ，则  $A[i]$  出来，否则  $A[j]$  出来。
- 向内找到第一个不相等的位置，就是找最远相等的位置，可以使用二分，与回文串类似，使用正反串上的 hash 值比较。
- 复杂度  $O(n \log n)$





1 例题 1

2 例题 2

3 例题 3

4 例题 4

5 例题 5

6 例题 6

7 例题 7

8 例题 8

9 例题 9



- 给你一个长度为  $n$  的字符串  $S$  和大小为  $t$  的字符串字典，有  $m$  个操作。
  - Q L R: 判断  $S$  从  $L$  到  $R$  的子串，在字典中是否出现过。
  - C X Y: 把  $S[X]$  修改为  $Y$ $n, m \leq 10^5, t \leq 10^4$   
(题目来源:HDU3973)



- 把字符串字典中的每个单词都 hash 成数值。



- 把字符串字典中的每个单词都 hash 成数值。
- 用线段树维护  $S$  的 hash 值。



- 把字符串字典中的每个单词都 hash 成数值。
- 用线段树维护  $S$  的 hash 值。
- 当前区间需要把左右两个儿子的 hash 值合并



- 把字符串字典中的每个单词都 hash 成数值。
- 用线段树维护  $S$  的 hash 值。
- 当前区间需要把左右两个儿子的 hash 值合并
- 设左区间的长度为  $lenL$ , hash 值为  $hL$   
右区间的长度为  $lenR$ , hash 值为  $hR$



- 把字符串字典中的每个单词都 hash 成数值。
- 用线段树维护  $S$  的 hash 值。
- 当前区间需要把左右两个儿子的 hash 值合并
- 设左区间的长度为  $lenL$ , hash 值为  $hL$   
右区间的长度为  $lenR$ , hash 值为  $hR$
- 当前区间的 hash 值为  $hL * Base[lenR] + hR$



1 例题 1

2 例题 2

3 例题 3

4 例题 4

5 例题 5

6 例题 6

7 例题 7

8 例题 8

9 例题 9





- 给出一个长为  $n$  由小写英文字母组成的字符串  $S$ ，再给出  $q$  个询问，要求回答  $S$  某个子串的最短循环节。  
如果字符串  $B$  是字符串  $A$  的循环节，那么  $A$  可以由  $B$  重复若干次得到。  
 $n \leq 5^5, q \leq 2 \times 10^6$   
(来源:POI2012 A Horrible Poem)



- 结论：如果  $A[1, n-x]$  与  $A[1+x, n]$  相等，则  $A$  有一个长度为  $x$  的循环节。



- 结论：如果  $A[1, n-x]$  与  $A[1+x, n]$  相等，则  $A$  有一个长度为  $x$  的循环节。
- 两个子串是否相等，可以用线段树维护 hash 值。



- 结论：如果  $A[1, n-x]$  与  $A[1+x, n]$  相等，则  $A$  有一个长度为  $x$  的循环节。
- 两个子串是否相等，可以用线段树维护 hash 值。
- 求  $S[L, R]$  的最短循环节，答案一定是  $m = R - L + 1$  的因子



- 结论：如果  $A[1, n-x]$  与  $A[1+x, n]$  相等，则  $A$  有一个长度为  $x$  的循环节。
- 两个子串是否相等，可以用线段树维护 hash 值。
- 求  $S[L, R]$  的最短循环节，答案一定是  $m = R - L + 1$  的因子
- 如果枚举  $m$  的所有因子，会超时。



- 结论：如果  $A[1, n-x]$  与  $A[1+x, n]$  相等，则  $A$  有一个长度为  $x$  的循环节。
- 两个子串是否相等，可以用线段树维护 hash 值。
- 求  $S[L, R]$  的最短循环节，答案一定是  $m = R - L + 1$  的因子
- 如果枚举  $m$  的所有因子，会超时。
- 考虑把  $m$  进行质因子分解， $m = p_1^{t_1} * p_2^{t_2} * \dots * p_k^{t_k}$ ，答案就是算出最终每个质因子的指数。



- 结论：如果  $A[1, n-x]$  与  $A[1+x, n]$  相等，则  $A$  有一个长度为  $x$  的循环节。
- 两个子串是否相等，可以用线段树维护 hash 值。
- 求  $S[L, R]$  的最短循环节，答案一定是  $m = R - L + 1$  的因子
- 如果枚举  $m$  的所有因子，会超时。
- 考虑把  $m$  进行质因子分解， $m = p_1^{t_1} * p_2^{t_2} * \dots * p_k^{t_k}$ ，答案就是算出最终每个质因子的指数。
- 如果循环节的长度为  $x$ ，则  $k * x$  也是循环节。



- 结论：如果  $A[1, n-x]$  与  $A[1+x, n]$  相等，则  $A$  有一个长度为  $x$  的循环节。
- 两个子串是否相等，可以用线段树维护 hash 值。
- 求  $S[L, R]$  的最短循环节，答案一定是  $m = R - L + 1$  的因子
- 如果枚举  $m$  的所有因子，会超时。
- 考虑把  $m$  进行质因子分解， $m = p_1^{t_1} * p_2^{t_2} * \dots * p_k^{t_k}$ ，答案就是算出最终每个质因子的指数。
- 如果循环节的长度为  $x$ ，则  $k * x$  也是循环节。
- 可以把每个指数依次变小，判断是否合法，如果某个指数减小 1 后，仍然合法，就减小 1。如果不合法，就调整下一个质因子的指数。复杂度小很多，不需要枚举所有的因子。

