

C++知识点总结(7): 枚举算法之最大公约数和最小公倍数

原创 AICodeThunder 于 2023-12-02 20:06:55 发布 阅读量249 收藏3 点赞数

分类专栏: C++知识点总结 文章标签: C/C++ 编程笔记 知识点总结 算法

C++知识点总结 专栏收录该内容

7 订阅 53 篇文章

一、枚举算法

枚举算法，将问题的所有可能的情况进行逐一列举，然后筛选出符合要求的一种程序处理算法。

枚举算法（特别是暴力枚举的时候）的缺点是，容易超时。一个计算机一般1秒最多运行1e8次，一旦超过1e8组数据，就有可能超时

枚举三要素：

- 1. 枚举对象（要枚举的对象）
- 2. 枚举范围（每一个枚举对象从几开始，到几结束）
- 3. 筛选条件（筛选满足一定条件的数据）



CSDN @AIC

二、最大公约数

约数：如果一个整数a能被整数b整除，那么b就是a的约数。

公约数：两个或者多个数公有的约数。

计算两个整数a和b的最大公约数，如何利用程序实现？

- 1. 枚举对象：1个数x（可能是最大公约数的数）
 - 2. 枚举范围：1 <= x <= min(a, b)
 - 3. 筛选条件：如果a % x是0，并且b % x也是0。
- 建议倒序遍历。

根据上述思路，我们写出代码：

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
```

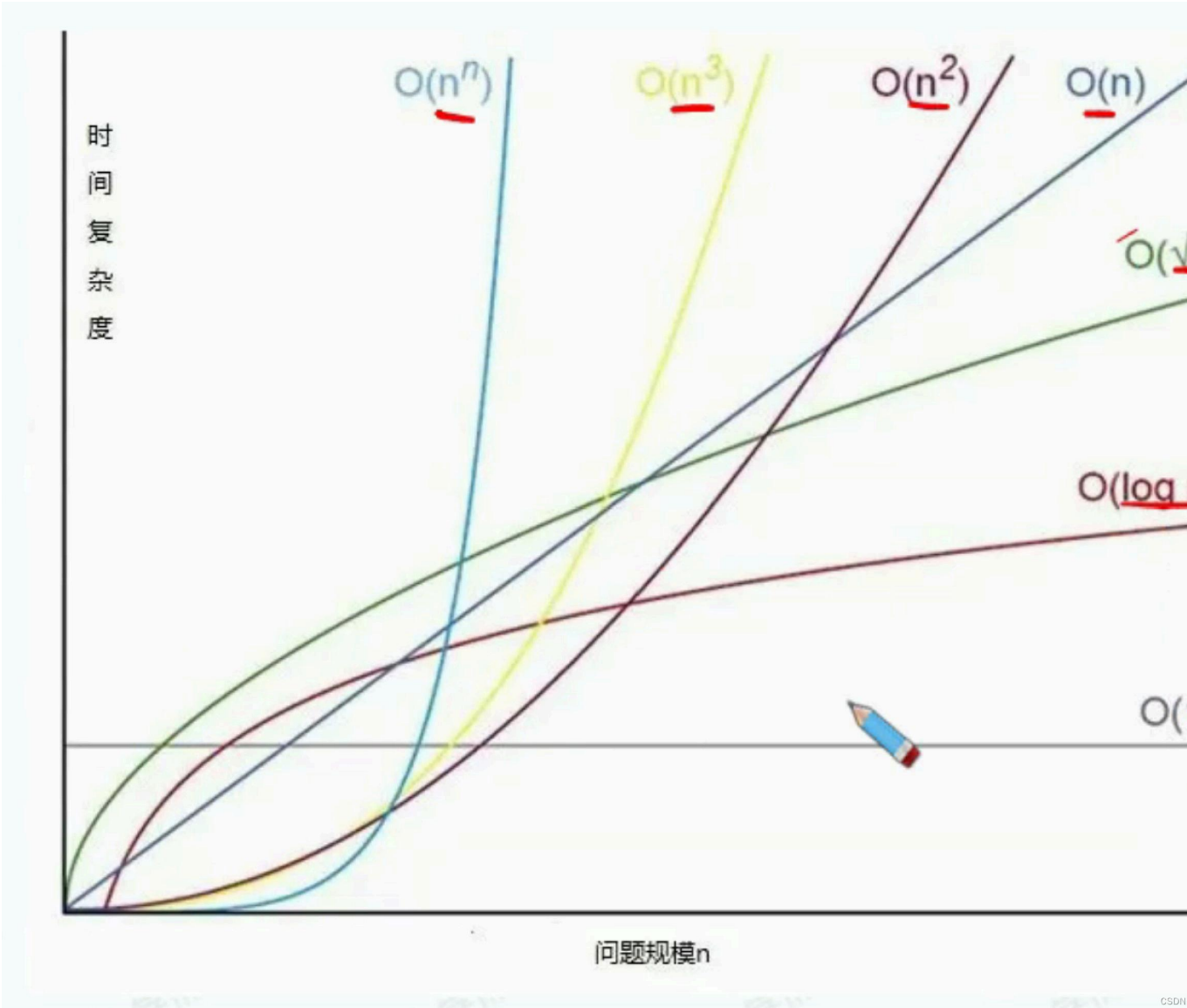
AICodeThunder 关注

0 3 0

```
5 { 6 | // 输入两个数字
7   int a, b;
8   cin >> a >> b;
9
10  // 枚举算法
11  int minn = min(a, b);
12  for (int i = minn; i >= 1; i--)
13  {
14      if (a % i == 0 && b % i == 0)
15      {
16          cout << i;
17          break;
18      }
19  }
20  return 0;
21 }
```



时间复杂度O(n), 概念图如下:



测试结果:



AICodeThunder

关注

0



3



0

CPU time limited exceeded



于是，我们还需要继续……嗯，现在教大家一种方法——**辗转相除法**，用了**就无敌了**！

将除数 b 当作下一次的被除数，余数 r 当作下一次的除数。如此反复地进行，一旦余数是 0 ，最后余数是 0 算式的除数。



比如说 $63 \div 24$ 。

$63 \div 24 = \text{余}15$

$24 \div 15 = \text{余}9$

$15 \div 9 = \text{余}6$

$9 \div 6 = \text{余}3$

$6 \div 3 = \text{余}0$

所以 63 和 24 的最大公约数是 3 。

辗转相除法程序逻辑：

```
1 int a, b, r;
2 while (a % b)
3 {
4     r = a % b; // 得到余数
5     a = b; // 除数作为下一次的被除数
6     b = r; // 余数作为下一次的除数
7 }
8 cout << b;
```

是的，这样就 OK 啦。

判题完成

1

✓

2

✓

3

✓

4

✓

5

✓

6

✓

7

✓

8

✓

9

✓

10

✓

Finish

AI Code Thunder

关注

0

3

0

三、最小公倍数

1. 枚举对象: 1 个数 y (可能是最小公倍数的数)
 2. 枚举范围: $\max(a, b) \leq y \leq a \times b$
 3. 筛选条件: 如果 $y \% a$ 是 0 , 并且 $y \% b$ 也是 0 。
- 建议正序遍历。

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      // 输入
7      int a, b;
8      cin >> a >> b;
9
10     // 枚举算法
11     int maxn = max(a, b);
12     for (int i = maxn; i <= a*b; i++)
13     {
14         if (i % a == 0 && i % b == 0)
15         {
16             cout << i;
17             break;
18         }
19     }
20     return 0;
21 }
```

拓展一个特殊关系:

整数 $a \times$ 整数 $b =$ 最大公约数 \times 最小公倍数

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      // 输入
7      long long a, b;
8      cin >> a >> b;
9      long long olda = a, oldb = b;
10
11     // 枚举算法
12     int r;
13     while (a % b)
14     {
15         r = a % b;
16         a = b;
17         b = r;
18     }
19     cout << olda*oldb/b;
20     return 0;
21 }
```

这样, 其实我们最小公倍数用的就是公式, 大部分都是最大公约数的程序。

四、真题

题目描述

输入两个正整数 $x, y(2 \leq x < 100000, 2 \leq y < 10000$
条件:



AICodeThunder

关注

👍 0 🗨 0 ⭐ 3 💬 0