D - 9 Divisors 解説

実行時間制限: 2 sec / メモリ制限: 1024 MB

配点 : 400 点

問題文

N 以下の正整数のうち、正の約数をちょうど 9 個持つものの個数を求めてください。

制約

- $1 \le N \le 4 \times 10^{12}$
- 入力される数値は全て整数

入力

入力は以下の形式で標準入力から与えられる。

```
N
```

出力

答えを出力せよ。

公式

D - 9 Divisors 解説 by nok0

整数 M が素数 $p_1, \dots, p_k$ を用いて $p_1^{r_1} \times p_2^{r_2} \times \dots \times p_k^{r_k}$ と素因数分解されるとき、M の約数の個数は $(r_1+1)(r_2+1)\dots(r_k+1)$ となることが知られています。

この事実を用いると、素数 p を用いて p^8 と表される正整数のうち N 以下のものの個数と、異なる素数 p, q を用いて $p^2 q^2$ と表される正整数のうち N 以下のものの個数が分かればよいです。

まず、前者は簡単です。素数を小さい順に見ていき、8 乗が N を超えたところで打ち切ればよいです。

後者について考えます。まず、p, q については $O(\sqrt{N})$ 以下の素数のみ考えればよいです。

p を昇順に全探索していきます。このとき、$p^2 q^2 \le N$ なる q の最大値は単調に減少していくので、尺取り法を用いることで、$O(\sqrt{N})$ 以下の素数の個数に対して線形時間でこの問題を解くことが出来ます。

投稿日時: 約 6 日前
最終更新: 約 1 日前

整数的素因数分解结论：

设一个正整数n的素因数分解式为：

$$n = p_1^{a_1} p_2^{a_2} \cdots p_r^{a_r}$$

其中 p_1, p_2, p_3 是不同的素数. $a_1, a_2, a_3, \ldots, a_r$ 分别是 $p_1, p_2, p_3, \ldots, p_r$ 出现的次数, 且都大于1。

则可以得到正整数 n 的因数的个数为:

$$\tau(n) = (a_1+1)(a_2+1)\cdots(a_r+1)$$

比如，

则72的因数的个数就是

$$\tau(72) = (3+1)(2+1) = 12$$

微信号: sx100sy

具体来说，我们可以看一看72的因数有哪些，数一数是不是12个，

1, 2, 3, 4, 6, 8, 9, 12, 18, 24, 36, 72

微信号: sx100sy

由以上结论可以知道，因数个数为9个的正整数为:

p^8 或 $p_1^2 p_2^2$

$(8+1)=9$

$(2+1)(2+1)=9$

所以只需要求出素数 $p^8 <= n$, 或者 $p_1^2 p_2^2 <= n$

由此将解法转为求正整数n的素数。

求质数使用埃拉托斯特尼筛法

go

```go
func eratosthenes(n int) []int {
	res := make([]int, 0)
	//isPrime是否是素数，0-是，1-否
	isPrime := make([]int, n)

	for i := 2; i < n; i++ {
		if isPrime[i] == 0 {
			res = append(res, i)
			for j := 2; j <= n; j++ {
				if i*j >= n {
					break
				}
				isPrime[i*j] = 1
			}
		}
	}
	return res
}
```

标记素数

标记由素数组成的合数

完整解法：

```go
func countNumbers(n int) int {
	c := 0
	limit := math.Sqrt(float64(n))

	prime := make([]int, int(limit)+1)

	for i := 1; i <= int(limit); i++ {
		prime[i] = i
	}

	for i := 2; i <= int(limit); i++ {
		if prime[i] == i {
			for j := i * i; j <= int(limit); j += i {
				if prime[j] == j {
					prime[j] = i
				}
			}
		}
	}

	for i := 2; i <= int(limit); i++ {
		p := prime[i]
		q := prime[i/prime[i]]

		if p*q == i && q != 1 && p != q {
			c++
		} else if prime[i] == i {
			if math.Pow(float64(i), 8) <= float64(n) {
				c++
			}
		}
	}

	return c
}
```

寻找N以内的素数时，若找到了一个大于√N的素数，则剩余的所有尚未标记的数也都是素数。

比如 i=2：4.6.8.10……
i=3：9.12.15.18……
标记这些数为非素数

标记合数的因数

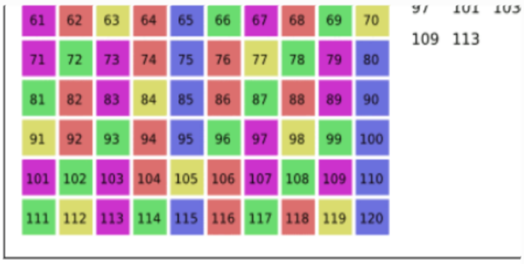
当prime[i]>i时，除数为不确定的数，不需用来判断是否为素数
反之除数为i本身

判断p,q是否素数

排除1和P的情况

当p=q，且prime[i]为质数时

埃拉托斯特尼篩法通過不斷地標記當前質數的所有倍數為合數，從而取得最小的未標記整數為下一個質數。不過，在實際使用此篩法尋找一個範圍內的質數時，不需要檢查範圍內所有整數，也不需要對每個質數都標記其所有的倍數。



使用埃拉托斯特尼篩法找出120以內的所有質數。由於11^2=121>120，當11成為最小的未標記整數時，尚未標記數皆可確認為質數。請注意到在標記時直接從每個質數的始。

1. 尋找N以內的質數時，若找到了一個大於$\sqrt{N}$的質數，則剩餘的所有尚未標記的數也都是質數。

   證明：若這些尚未標記的數中有任意一個為合數，設之為m，則m必定是除1與自身以外的兩個因數的乘積。但既然m尚未被標記，則所有小於等於$\sqrt{N}$的數均不是m的因數。故這兩個因數必然都大於$\sqrt{N}$，則m不可能在N以內[3]:103-104[4]:4。

2. 標記某一質數p的倍數時，不需要每次皆從$2p, 3p, ...$開始，而可直接從p^2開始標記。

   證明：所有較p^2更小的p的倍數必然擁有一個更小的質數為其因數，故在標記之前的質數的倍數時它們已經被標記過了[5]。

若要找出25以內的所有質數，使用如上述改進過的埃拉托斯特尼篩法的具體過程如下：

1. 列出2以後所有數：

   2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

2. 記錄質數2，由2^2=4開始划去2的倍數：

   2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 ~~16~~ 17 ~~18~~ 19 ~~20~~ 21 ~~22~~ 23 ~~24~~ 25

3. 記錄下一質數3，由3^2=9開始划去3的倍數：

   2 3 ~~4~~ 5 ~~6~~ 7 ~~8 9 10~~ 11 ~~12~~ 13 ~~14 15 16~~ 17 ~~18~~ 19 ~~20 21 22~~ 23 ~~24~~ 25

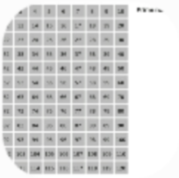
4. 記錄下一質數5，由5^2=25開始划去5的倍數：

   2 3 ~~4~~ 5 ~~6~~ 7 ~~8 9 10~~ 11 ~~12~~ 13 ~~14 15 16~~ 17 ~~18~~ 19 ~~20 21 22~~ 23 ~~24 25~~

5. 下一質數為7，而7^2=49>25，故剩餘所有未標記的數皆為質數：

   2 3 ~~4~~ 5 ~~6~~ 7 ~~8 9 10~~ 11 ~~12~~ 13 ~~14 15 16~~ 17 ~~18~~ 19 ~~20 21 22~~ 23 ~~24 25~~

由此得到25內的質數為2，3，5，7，11，13，17，19，23。



埃拉托斯特尼篩法 - 維基百科，自由的百科...

埃拉托斯特尼篩法 - 維基百科，自由的百科全書 首頁...

zh.m.wikipedia.org