



# 质因数分解

## 方案一

枚举从 2 到  $a$  的每个数，如果  $i$  是  $a$  的一个因数，尝试把他除去并累计幂次。

```
for(int i=2;i<=a;i++)while(a%i==0)a/=i,mp[i]++; //分解质因数
```

## 方案二

在方案一的基础上进行小优化，注意到：

” 引理

如果  $a$  有大于  $\sqrt{a}$  的质因数，那么它一定是唯一的。



证明

反证法，如果有两个大于  $\sqrt{a}$  的质因数，相乘后一定大于  $a$ 。

```
for(int i=2; i*i<=a; i++) {
    while(a%i == 0) {
        a /= i;
        mp[i]++;
    }
}
if(a > 1) mp[a]++;
// 处理最后剩下的质因数
```

## 方案三

先打质数表再比较，比较复杂但是快一些。

以下段落为 AI 辅助生成，不保证正确性。

```
const int MAXN = 50000000;
bool is_prime[MAXN + 1];
vector<int> primes;
void sieve() {
    fill(is_prime, is_prime + MAXN + 1, true);
    for (int i = 2; i * i <= MAXN; i++) {
        if (is_prime[i]) {
            for (int j = i * i; j <= MAXN; j += i)
                is_prime[j] = false;
        }
    }
    for (int i = 2; i <= MAXN; i++)
        if (is_prime[i])
            primes.push_back(i);
}
```

```
is_prime[0] = is_prime[1] = false;
for (int i = 2; i * i <= MAXN; ++i) {
    if (is_prime[i]) {
        for (int j = i * i; j <= MAXN; j += i) {
            is_prime[j] = false;
        }
    }
}
for (int i = 2; i <= MAXN; ++i) {
    if (is_prime[i]) {
        primes.push_back(i);
    }
}
}
```

---

然后用质数表进行分解：

---

以下段落为 AI 辅助生成，不保证正确性。

```
for (int prime : primes) {
    if (prime * prime > a) break;
    while (a % prime == 0) {
        a /= prime;
        mp[prime]++;
    }
}
if (a > 1) mp[a]++;
}
```