

Рассмотрим следующую функцию, которая раскладывает целое число x на простые множители. Пусть число x не превышает C . Оцените асимптотику времени работы этой функции.

C++:

```
1 #include <vector>
2
3 std::vector<int> primes(int x) {
4     std::vector<int> result;
5     int i = 2;
6     while (i * i <= x) {
7         while (x % i == 0) {
8             result.push_back(i);
9             x /= i;
10        }
11        ++i;
12    }
13    if (x != 1) {
14        result.push_back(x);
15    }
16    return result;
17 }
```

Python:

```
1 def primes(x):
2     result = []
3     i = 2
4     while i * i <= x:
5         while x % i == 0:
6             result.append(i)
7             x //= i
8         i += 1
9     if x != 1:
10        result.append(x)
11    return result
```

-
- $O(\sqrt{C} + \log C)$
 - $O(\sqrt{C} \log C)$
 - $O(\log C)$
 - **$O(\sqrt{C})$**
 - $O(C)$

В худшем случае алгоритм не заходит во внутренний цикл и итерируется во внешнем \sqrt{n} раз. В общем случае x уменьшается логарифмически, сокращая сложность.

