Consider the following function, which decomposes the integer x into prime factors. Let the number x not exceed C. Estimate the asymptotic behavior of the running time of this function.

C++:

```cpp
#include <vector>

std::vector<int> primes(int x) {
    std::vector<int> result;
    int i = 2;
    while (i * i <= x) {
        while (x % i == 0) {
            result.push_back(i);
            x /= i;
        }
        ++i;
    }
    if (x != 1) {
        result.push_back(x);
    }
    return result;
}
```

Python:

```python
def primes(x):
    result = []
    i = 2
    while i * i <= x:
        while x % i == 0:
            result.append(i)
            x //= i
        i += 1
    if x != 1:
        result.append(x)
    return result
```

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

In the worst case, the algorithm does not go into the inner loop and iterates in the outer $\sqrt{C}$ times. In general, x decreases logarithmically, reducing complexity.