**Нахождение делителей числа**

В зависимости от задачи можно выделить два подхода для нахождения делителей числа n:

* нахождение абсолютно всех делителей;
* нахождение всех простых делителей и получение из них нужных делителей.

Реализация первого подхода (сложность – О(sqrt(n)), type – int или long long):

for (type i = 1; i \* i <= n; i++)

{

if (n % i == 0)

{

// нашли 2 новых делителя числа: i и (n/i)

// производим нужные действия с этими двумя найденными делителями

}

}

Реализация второго подхода (сложность – O(sqrt(n)\*log(n)), type – int или long long):

map<type, type> primeFactorsCount;

for (type i = 2; i \* i <= n; i++)

{

while (n % i == 0)

{

if (primeFactorsCount.count(i) == 0)

{

primeFactorsCount[i] = 0;

}

primeFactorsCount[i]++;

n /= i;

}

}

if (n != 1)

{

if (primeFactorsCount.count(n) == 0)

{

primeFactorsCount[n] = 0;

}

primeFactorsCount[n]++;

}

По окончании работы алгоритма имеем словарь, где ключами являются простые делители числа, а значениями – сколько раз они встречаются в этом числе (в какой степени). Зная это, можно получить любой делитель числа, выбрав для каждого простого делителя степень от 0 до значения в словаре и перемножить их.