Отчет по лабораторной работе № 4. Вычисление наибольшего общего делителя

дисциплина: Математические основы защиты информации и информационной безопасности

Наливайко Сергей Максимович

Содержание

# Цель работы

Научиться реализовывать алгоритмы нахождение НОД.

# Задание

* Реализовать алгоритм Евклида нахождения НОД
* Реализовать бинарный алгоритм Евклида нахождения НОД
* Реализовать расширенный алгоритм Евклида нахождения НОД
* Реализовать расширенный бинарный алгоритм Евклида нахождения НОД

# Выполнение лабораторной работы

## Реализация алгоритмов нахождения НОД

Реализуем алгоритмы нахождения НОД на языке программирования C++.

Код функции нахождения НОД для первого алгоритма:

int euclidean\_algorithm(int a, int b) {  
 return b == 0 ? a : euclidean\_algorithm(b, a % b);  
}

Код функции нахождения НОД для второго алгоритма:

int bin\_euclidean\_algorithm(int a, int b) {  
 if (b <= 0 || a <= 0)  
 throw std::invalid\_argument("numbers must be greater than zero");  
 if(a < b)  
 std::swap(a, b);  
 int g = 1;  
 while ((a % 2 == 0) && ( b % 2 == 0))  
 {  
 a /= 2;  
 b /= 2;  
 g \*= 2;  
 }  
 int u = a, v = b;  
 while (u != 0)  
 {  
 while (u % 2 == 0) u /= 2;  
 while (v % 2 == 0) v /= 2;  
 if (u >= v)  
 u -= v;  
 else  
 v -= u;  
 }  
 return v \* g;  
}

Код функции нахождения НОД для третьего алгоритма:

int ext\_euclidean\_algorithm(int a, int b, int &x, int &y) {  
 if (b < 0 || a < 0)  
 throw std::invalid\_argument("numbers must be greater than zero");  
 if(a < b)  
 std::swap(a, b);  
  
 int s, d;  
  
 if (b == 0) {  
 d = a;  
 x = 1;  
 y = 0;  
 return d;  
 }  
  
 d = ext\_euclidean\_algorithm(b,a % b,x,y);  
 s = y;  
 y = x - (a / b) \* (y);  
 x = s;  
 return d;  
}

Код функции нахождения НОД для четвертого алгоритма:

int ext\_bin\_euclidean\_algorithm(int a, int b, int &x, int &y) {  
 if (b <= 0 || a <= 0)  
 throw std::invalid\_argument("numbers must be greater than zero");  
 if(a < b)  
 std::swap(a, b);  
  
 int g = 1;  
 while ((a % 2 == 0) && ( b % 2 == 0))  
 {  
 a /= 2;  
 b /= 2;  
 g \*= 2;  
 }  
 int u = a, v = b, A = 1, B = 0, C = 0, D = 1;  
 while (u != 0)  
 {  
 while (u % 2 == 0) {  
 u /= 2;  
  
 if ((A % 2 == 0) && (B % 2 == 0)) {  
 A /= 2;  
 B /= 2;  
 } else {  
 A = (A + b) / 2;  
 B = (B - a) / 2;  
 }  
 }  
 while (v % 2 == 0) {  
 v /= 2;  
  
 if ((C % 2 == 0) && (D % 2 == 0)) {  
 C /= 2;  
 D /= 2;  
 } else {  
 C = (C + b) / 2;  
 D = (D - a) / 2;  
 }  
 }  
 if (u >= v) {  
 u -= v;  
 A -= C;  
 B -= D;  
 }  
 else {  
 v -= u;  
 C -= A;  
 D -= B;  
 }  
 }  
 x = C;  
 y = D;  
 return v \* g;  
}

Полный листинг программного кода представлен в файле main.cpp (архив lab04, директория src).

Скомпилируем и запустим программу fig. 1.

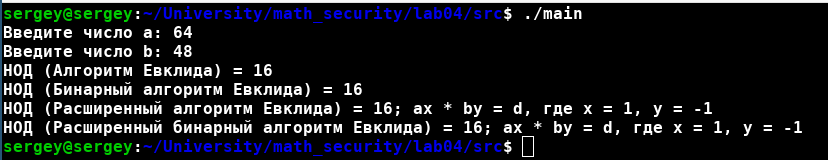


Figure 1: Нахождение НОД различными алгоритмами

# Выводы

В ходе лабораторной работы мы реализовывали алгоритмы нахождение НОД.