CS211: Algorithms & Data structures

Dr. Sameer M. Alrehaili

September 15, 2021

srehaili@taibahu.edu.sa
college of computer science and engineering ,yanbu, Taibah University

1 Laboratory Objectives:

- To understand that there are a number of ways of solving one problem.
- To identify which way is most eficient one.

2 Exercises

We will take the problem of finding the Greatest Common Divisor (GCD). Three different solutions will be applied in this lab and experimental method is going to be used to decided which one of them is the most efficient.

The Greatest Common Divisor (GCD) of two integers, a and b, is the largest positive integer number that divides both of them without leaving any reminder. For example, 6 is the GCD of 12 and 18

2.1 Brute Force Method

Find the greatest common divisor (GCD) of two integers, a and b.

Algorithm 1: GCD

Input: Two integer numbers a and b
Output: *gcd*

1: m $\leftarrow$ The minimum number of a and b.
2: gcd $\leftarrow 0$
3: i $\leftarrow 2$
4: **while** $(i \leq m)$ **do**
5: **if** $a \ mod \ i = 0$ and $b \ mod \ i = 0$ **then**
6: $gcd \leftarrow i$
7: **end if**
8: i $\leftarrow i + 1$
9: **end while**
10: **return** *gcd*

?? Brute Force method loops for 48 times to find gcd(80, 48), while recursive method cost 4 iteration and subtraction method only cost 3 iterations. This algorithm is inefficient and tedious if a and b are big and the gcd of the two numbers is equal to one.

Listing 1: GCD

```
// Hello.java
public static int GCD1_1(int a, int b){
    int m = Math.min(a,b);
    int gcd=0;
    int i=2;
    while(i<=m)
    {
        if(a%i ==0 && b%i==0)
            gcd=i;
    i++;
    }
    return gcd;
}
```

2.2 Subtraction Method

Algorithm 2: GCD

Input: Two integer numbers a and b
Output: a

```
1: while (a ≠ b) do
2:     if a > b then
3:         a ← a − b
4:     else
5:         b ← b − a
6:     end if
7: end while
8: return a
```

2.3 Euclid's algorithm using iterative method

$gcd(a, b) = gcd(b_i, a_i \bmod b_i)d$
$ax_0 + by_0 = d$
$a_{i+1} = b_i$
$b_{i+1} = a_i \bmod b_i$

Algorithm 3: GCD

Input: Two integer numbers a and b
Output: gcd

```
1: while (b ≠ 0) do
2:     r ← a mod b
3:     a ← b
4:     b ← r
5: end while
6: return gcd
```

1. Write the three algorithms mentioned above and calculate elapsed time for each one for the same set of inputs?