Clay Buxton

Homework 1

CS322 Algorithms
January 22, 2020

Problem 1

Prove gcd(m,n) | gcd(n,m%n)

Let $c = \gcd(n,m\%n)$ for all integers n,m where $n > m$
Therefore $c|n$ and $c|m\%n$
If $c|m\%n$ then $c|km+n$ for some integer k
if $c|km+n$ then $c|\gcd(m,n)$ by definition of gcd

Problem 2

Implement the Sieve algorithm *Also see primes.cpp*

Algorithm 1: Sieve Algorithm

```cpp
std::vector<int> primes::sieve(int n){
    //Initlize List
    std::vector<int> primes;
    for(int i = 0; i < n; i++){
        primes.push_back(i);
    }
    for (int i = 2; i <= n; i++){
        if(!std::count(primes.begin(),primes.end(),i){
            continue;
        }
        for (int j = i + i; j <= n; j += i){
            primes.erase(std::remove(primes.begin(), primes.end(), j), primes.end(
        }
    }
    return primes;
}
```

Problem 3

1. **Implement the Extended Euclidean Algorithm** *Also see euclidean.cpp*

Algorithm 2: Extended Euclidean Algorithm

```
int euclidean::gcdExtended(int a, int b, int *x, int *y){
    if (a == 0){
        *x = 0;
        *y = 1;
        return b;
    }
    int x1, y1;
    int gcd = gcdExtended(b%a, a, &x1, &y1);

    *x = y1 - (b/a) * x1;
    *y = x1;

    return gcd;
}
```

2. **Find the integer solution to the Diophantine problem** *Also see euclidean.cpp*

Algorithm 3: Diophantine Equation

```
void euclidean::diophantine(int a, int b, int c, int *x, int *y) {
    int gcd = gcdExtended(a,b,x,y);
    if(c%gcd){
        std::cout << "No Solution" << std::endl;
    }
    int d = c / gcd;
    *x *= d;
    *y *= d;
    return;
}
```

Problem 4

1. 14,25,47,60,81,98
2. No, if given two numbers of the same value the algorithm will not work.
3. No, it uses 2 arrays the "Count" and "A" arrays, and then needs another array to output the values to.