

CS473 ASSIGNMENT-1

Boran Yildirim Yigit Bediskan

February 2018

1

- (a) **Loop Invariant:** $\gcd(a, b) = \gcd(x, y)$ and $x \geq 0$ and $y \geq 0$
- (b) **Initialization:** At the beginning of the first iteration x is still equal to a , and y is still equal to b , therefore,

$$\gcd(a, b) = \gcd(x, y) \quad \text{and} \quad x \geq 0 \quad \text{and} \quad y \geq 0$$

Maintenance: Assume that at the beginning of the i th iteration,

$$\gcd(a, b) = \gcd(x, y) \quad \text{and} \quad x \geq 0 \quad \text{and} \quad y \geq 0$$

In the i th iteration, if $(x > y)$, then $x_1 = x - y$, x_1 is positive, and $\gcd(x_1, y) = \gcd(a, b)$, because any number that divides x and y also divides $x - y$, else if $(x < y)$, then $y_1 = y - x$, y_1 is positive, and $\gcd(x, y_1) = \gcd(a, b)$, because any number that divides x and y also divides $y - x$.

- (c) **Termination:** The loop terminates when $x = y$. At this point, since $\gcd(a, b) = \gcd(x, y)$ and $x \geq 0$ and $y \geq 0$ Thus the procedure returns x .

2

(a)

```
Require: A, B, n  
   $i \leftarrow 0$   
   $j \leftarrow n - 1$   
  while  $i < n$  and  $j > 0$  do  
     $tmp \leftarrow A[i] + B[j]$   
    if  $tmp == x$  then  
      return FOUND  
    else if  $tmp > x$  then  
       $j \leftarrow j - 1$   
    else if  $tmp < x$  then  
       $i \leftarrow i + 1$   
    end if  
  end while  
  
  return NOTFOUND
```

(b) It is attached at the end of the report and handled with worst case together.

(c)

(d) It is attached at the end of the report and handled with explanation of how algorithm work.

3

Model: MacBook Pro (Retina, 15-inch, Mid 2015)

CPU: 2,2 GHz Intel Core i7

RAM: 16 GB 1600 MHz DDR3

Operating System: macOS High Sierra

- (a) The arrays are randomly generated for both of them with same values. After the size of array become greater than 2^8 , Merge-Sort become more efficient than Insertion-Sort. Merge-Sort has a more variable run time for different instances that have the same size since it has $O(2n)$ space complexity where Insertion-Sort has $O(n)$.

n	Insertion Sort	Merge Sort
2^4	1e-06	6e-06
2^8	4.9e-05	3.4e-05
2^{12}	0.012053	0.0006336
2^{16}	3.03937	0.012187
2^{20}	∞	0.259265

- (b) The worst case for Insertion-Sort is with numbers in descending order in that case. The critical value of n in this case is approximately 2^6 .

n	Insertion Sort	Merge Sort
2^4	1e-06	6e-06
2^8	9.7e-05	2.2e-05
2^{12}	0.028046	0.00036
2^{16}	6.31218	0.009998
2^{20}	∞	0.138445

Plots of the runtime of each algorithm are attached at the end of report.