

# University of Nevada, Las Vegas Computer Science 477/677 Spring 2020

## Assignment 3: Due Thursday February 20, 2020

Name: \_\_\_\_\_

You are permitted to work in groups, get help from others, read books, and use the internet. But the handwriting on this document must be your own. Print out the document, staple, and fill in the answers. You may attach extra sheets. Turn in the pages to the graduate assistant at the beginning of class, August 14.

1. Give the asymptotic time complexity in terms of  $n$ , using  $\Theta$ ,  $O$ , or  $\Omega$ , whichever is most appropriate.

(a)  $F(n) \geq F(n - \sqrt{n}) + n^2$

(b)  $H(n) < H(n/3) + H(n/4) + 2H(n/5) + n$

(c)  $G(n) = 3(G(2n/3) + G(n/3)) + 5n^2$

2. For each of these recursive subprograms, write a recurrence for the time complexity, then solve that recurrence.

(a) 

```
void george(int n)
{if(n > 0)
  {for(int i = 0; i < n; i++) cout << "hello" << endl;
  george(n/2); george(n/3); george(n/6);}}
```

(b) 

```
void martha(int n)
{if (n > 1)
  {martha(n-1); martha(n-2):}}
```

Hint: Look at problem 0.3 on page 9 of your textbook.

3. The following function computes  $ab$  for positive integers  $a$  and  $b$ . The loop invariant for this code is  $p + cd = ab$ .

```
int product(int a, int b)
{
    int c = a;
    int d = b;
    int p = 0;
    while(d > 0)
    {
        if(d % 2) p = p+c;
        c = c+c;
        d = d/2;
    }
    return p;
}
```

The following function computes  $x^b$  for a real number  $x$  and a positive integer  $b$ . What is its loop invariant?

```
float power(float x, int b)
{
    float y = x;
    int d = b;
    float z = 1.0;
    while(d > 0)
    {
        if(d % 2) z = z*y;
        y = y*y;
        d = d/2;
    }
    return z;
}
```

4. Consider the following program, where  $n$  is a given constant.

```
int A[n];
int B[n];
void getA()
{
    for(int i = 0; i < n; i++) cin >> A[i];
}
int main()
{
    getA();
    int s = 0;
    int i = 0;
    while(i < n) // beginning of outer loop
    {
        while(s > 0 and B[s-1] > A[i]) s--; // inner loop
        B[s] = A[i];
        s++;
        i++;
    } // end of outer loop
    for(int j = 0; j < s; j++) cout << B[j];
    cout << endl;
    return 1;
}
```

- (a) If  $n = 10$  and the input stream is 0 6 9 8 1 3 2 5 4 7 what is the output?
- (b) The inner and outer loops are both linearly bounded, and thus the time complexity of the code is  $O(n^2)$ . But, it is not  $\Theta(n^2)$ . Use amortization to prove that the time complexity is  $\Theta(n)$ .