

Data Structures and Programming

Spring 2017, Homework # 1

Due: April 11, 2017

1. (20 pts) List the functions below from lowest to highest order. If any two (or more) are of the same order, indicate which. (You do not have to formally justify each relation.) Note: $\lg n = \log_2 n$ and $\ln n = \log_e n$.

- n
- 2^n
- n^k , where $k > 5$
- $n(\lg n)$
- $\ln n$
- 192
- $n - n^3 + 7n^5$
- $\lg n$
- \sqrt{n}
- $\lg \lg n$
- $O(1)$
- n^3
- $(\lg n)^2$
- $n!$
- $n^{1+\epsilon}$, where $0 < \epsilon < 1$.

2. (30 points) For each of the following program fragments, give a big-Oh analysis of the running time. You can use a rule-of-thumb approach.

```
// Fragment #1
for (int i = 0; i < m; i + = 2)
    sum++;
```

```
// Fragment #2
for (int i = 0; i < n; i ++ )
    for (int j = 0; j < m; j ++ )
        sum++;
```

```
// Fragment #3
for (int i = 0; i < n; i ++ )
    sum++;
for (int j = 0; j < n; j ++ )
    sum++;
```

```
// Fragment #4
for (int i = 0; i < n; i ++ )
    for (int j = 0; j < i; j ++ )
        sum++;
```

```
// Fragment #5
for (int i = 0; i < n; i ++ )
    for (int j = 0; j < i; j ++ )
        for (int k = 0; k < j; k ++ )
            sum++;
```

```
// Fragment #6
for (int i = 1; i < n; i = i * 2)
    sum++
```

3. (15 points) Convert the following infix expressions to postfix form by using the algorithm given in this chapter. Show the status of the stack after each step of the algorithm.

(a) $a/b/c - (d + e) * f$

(b) $a * (b/c/d) + e$

(c) $a - (b + c * d)/e$

4. (15 points) Consider the recursive algorithm for calculating x^n . Calculate how many multiplications happen for each value of $n = 1$ to $n = 40$. Using a plotting program (such as Excel), plot n versus the number of multiplications. For some constants a and b , does $a * \log(n) + b$ match the data? Is this what you expected? Why?

5. (20 points) The "hailstone procedure" is defined as follows. Start with a number X . If X is even, divide it by 2. If X is odd, compute $3X + 1$. Keep doing this until you reach 1.

Step 1. $X = 3$.

Step 2. $X = 3 * 3 + 1 = 10$.

Step 3. $X = 10/2 = 5$.

Step 4. $X = 3 * 5 + 1 = 16$.

Step 5. $X = 16/2 = 8$.

Step 6. $X = 8/2 = 4$.

Step 7. $X = 4/2 = 2$.

Step 8. $X = 2/2 = 1$.

```
public static int HailstoneNumber(int X) {
    int count = 1;
    while (X != 1) {
        if (X % 2 == 1) X = 3*X + 1;
        else X = X/2;
        count++;
    }
    return count;
}
```

Questions:

- (a) Rewrite this as a recursive function. The answer is short, and actually simple.
 (b) Find the hailstone number for each of the following numbers

3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19