

# EECE.3220: Data Structures

Spring 2017

## Homework 1

Due **1:50 PM, Friday, 2/8/17**

### Notes:

- While typed solutions are preferred, handwritten solutions are acceptable.
- All solutions must be legible and contained in one file. Archive files are not acceptable.
- Electronic submissions should be uploaded to your Dropbox folder. Please include your name as part of your filename (for example, mgeiger\_hw1.pdf).

1. (25 points) Assume each expression listed below represents the execution time of a program. Express the order of magnitude for each time using big O notation.

a.  $T(n) = n^3 + 100n \cdot \log_2 n + 5000$

b.  $T(n) = 2^n + n^{99} + 7$

c.  $T(n) = \frac{n^2-1}{n+1} + 8 \log_2 n$

d.  $T(n) = 1 + 2 + 4 + \dots + 2^{n-1}$

2. (75 points) For each of the code segments below, determine an equation for the worst-case computing time  $T(n)$  (expressed as a function of  $n$ , *i.e.*  $2n + 4$ ) and the order of magnitude (expressed using big O notation, *i.e.*  $O(n)$ ).

a. 

```
// Calculate mean
n = 0;
sum = 0;
cin >> x;
while (x != -999)
{
    n++;
    sum += x;
    cin >> x;
}
mean = sum / n;
```

2. (continued) (?? points) For each of the code segments below, determine an equation for the worst-case computing time  $T(n)$  (expressed as a function of  $n$ , i.e.  $2n + 4$ ) and the order of magnitude (expressed using big O notation, i.e.  $O(n)$ ).

- b. `// Matrix addition`  
`for (int i = 0; i < n; i++) {`  
    `for (int j = 0; j < n; j++) {`  
        `c[i][j] = a[i][j] + b[i][j];`  
    `}`  
`}`
- c. `// Matrix multiplication`  
`for (int i = 0; i < n; i++) {`  
    `for (int j = 0; j < n; j++) {`  
        `c[i][j] = 0;`  
        `for (int k = 0; k < n; k++) {`  
            `c[i][j] += a[i][k] * b[k][j];`  
        `}`  
    `}`  
`}`
- d. `// Bubble sort`  
`for (int i = 0; i < n - 1; i++) {`  
    `for (int j = 0; j < n - 1; j++) {`  
        `if (x[j] > x[j + 1]) {`  
            `temp = x[j];`  
            `x[j] = x[j + 1];`  
            `x[j + 1] = temp;`  
        `}`  
    `}`  
`}`
- e. `while (n >= 1)`  
    `n /= 2;`
- f. `x = 1;`  
`for (int i = 1; i <= n - 1; i++) {`  
    `for (int j = 1; j <= x; j++)`  
        `cout << j << endl;`  
    `x *= 2;`  
`}`