EECE.3220: Data Structures

Spring 2019

Homework 1

Due **11:59 PM**, **Wednesday, 2/20/19 (NO LATE SUBMISSIONS)**

Notes:

- All solutions to this assignment must be electronically submitted to Blackboard.
- You may handwrite your solutions and scan the pages, **but all solutions must be legible and contained in one file**. Archive files are *not* acceptable.
- With the potential for Exam 1 to be on Friday, 2/22, **I will not accept late submissions for this assignment so I can post the solution Thursday, 2/23**.
    - If the exam gets scheduled for the following week, I will relax this requirement and allow late submissions. I'll announce any changes on Blackboard and in class.

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

2. (75 points + 5 extra credit) 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) (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)).

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. *(extra credit—5 points)*
```
x = 1;
for (int i = 1; i <= n - 1; i++) {
   for (int j = 1; j <= x; j++)
      cout << j << endl;
   x *= 2;
}
```