COP 3502 Section 2: Algorithm Analysis Homework

Assigned: Friday, June 12, 2020
Due: Friday, June 19, 2020 (via Webcourses Section 2)

Directions: Please either type your answers in a .doc or .docx file or write on paper and scan your answers into a SINGLE .pdf file and submit that file over Webcourses. (Thus, the accepted file types will be .doc, .docx and .pdf.) Check Webcourses for the specific time the assignment is due.

Timing Problems

1) An algorithm with an $O(n^2)$ run time takes 24 ms to process input data with size n = 10000. How long will the algorithm take, in seconds, to process input data with size n = 50000?

2) An algorithm that takes in as input an array with n rows and m columns has a run time of O(nlgm). The algorithm takes 173 ms to run in an input array with 1000 rows and 512 columns. How long will the algorithm take to run on an input array with 1500 rows and 4096 columns? (Note: For ease of calculation, please use a base of 2 for your logarithm.)

3) An algorithm has a run time of $O(n^k)$ for some integer k. On an input of size 500, the algorithm takes 16 seconds to run. On an input of size 750, the algorithm takes 81 seconds to run. What is the value of k?

Summation Problems

4) What is the value of the following summation: $\sum_{i=30}^{75}(3i + 4)$?

5) What is the value of the following summation in terms of n: $\sum_{i=n+1}^{3n}(2i - 3)$?

6) Determine the value of this summation, in terms of a, b, c and d. Assume that a < b and that all four are positive integers. $\sum_{i=a}^{b}(ci + d)$.

Big-Oh Simplification

For each of these problems, take the given function and rewrite it as the most simple Big-Oh class the function belongs to:

7) $f(n) = 3n^{10} - 99n^9 + 3n + 2$

8) $f(n) = 3n\lg n + (.0001)n^2 + 20000$

9) $f(n) = \frac{n^3+3n^2+7}{2n^2-(n+7)(2n+3)+27n}$

10) $f(n) = \lg n + \lg n^2 + \lg(\lg n) + (lg^2 n)$

11) $f(n) = n^{1.999} + \frac{n^2}{\lg n}$

Code Segment Analysis Problems

Give a Big-Oh run time for each of the following functions, in terms of the variables in the problem. (These will typically be n and m, respectively, and some of the problems will only have 1 variable and others will have two.) Provide a succinct proof/reason for your answer.

12)

```
int f(int n, int m) {
    int sum = 0;
    for (int i=0; i<n; i++)
        sum++;
    for (int i=0; i<m; i++)
        sum++;
    return 0;
}
```

13)

```
int f(int n, int m) {
    int sum = 0;
    for (int i=0; i<n; i++)
        for (int j=0; j<m; j++)
            sum++;
    return sum;
}
```

14)

```
int f(int* array, int n) {
    int low = 0, high = n-1;
    while (low < high) {
        int mid = (low+high)/2;
        if (array[mid] < target)
            low = mid+1;
        else if (array[mid] > target)
            high = mid-1;
        else
            return 1;
    }
    return 0;
}
```

15)

```
int f(int** array, int n) {
    int x = 0, y = 0;
    while (x < n) {
        y = 0;
        while (y < n && array[x][y] != 0)
            y++;
        x++;
    }
    return y;
}
```

16)

```
int f(int** array, int n) {
    int x = 0, y = 0;
    while (x < n) {
        while (y < n && array[x][y] != 0)
            y++;
        x++;
    }
    return y;
}
```

17)

```
int f(int n) {
    int sum = 0;
    while (n > 0) {
        sum += (n%2);
        n /= 2;
    }
    return sum;
}
```