Assignment 3

Timothy Hoang

2/17/19

CMPT435-111

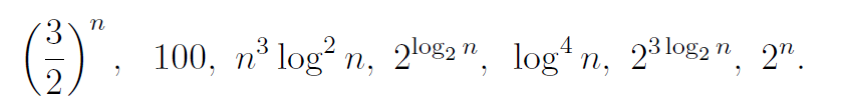
**Date Assigned: 02/11/2019**

**Due: Midnight 02/17/2019 on iLearn**

**Please read turn-in checklist at the end of this document before you start doing exercises.**

**Section 1: Pen-and-paper Exercises**

1. Arrange the following functions in ascending order of growth rate (7 points):



You are NOT required to justify your ordering.

Note:

In this problem, you are asked to identify if f1(n) < f2(n) for a “sufficiently large” input size n. However, for small values of n this is not always true.

(3/2)^n = n + log(3/2)

n^3 log^2 n = n^3 logn logn = (3+logn) loglogn loglogn

2^logn = logn + log2

log^4 n = logn logn logn logn = loglogn loglogn loglogn loglogn

2^3logn = 2logn + log2

2^n = n + log2

100, log^4 n, n^3 log^2 n, 2^logn, 2^3logn, (3/2)^n, 2^n

* 1. Analyze the following code and provide a "Big-O" estimate of its running time in terms of n. Explain your analysis. Assume k is a constant given by the problem.

for (i=1; i<=n; i++) //n times

p = pow(i,k); // p = i to the power of k //2

for (j=1; j<=p; j++) //n^k times

Some O(1) work

end for

end for

Note:

The value of p changes in each iteration of the outer loop, as a result, the upper bound for j changes as well.

Credit will not be given only for answers - show all your work:

(5 points) steps you took to get your answer.

First for loop runs n times

P = i^k

1^k, 2^k, 3^k, … n^k = n^k + n^k + n^k …(n times)… + n^k

= n\*n^k

n^(k+1)

Second for loop runs p times

In total it runs (n+1)^k times

N\*(2\*n^k)=n\*2n^k=2\*n^(k+1)

(5 points) your answer.

2\*n^(k+1)

1. For each of the following tasks, find the complexity of the algorithm using big O notation. You must justify your answer with 1-2 lines of explanation.
2. Insert a new element into an unsorted array.

Note: Credit will not be given only for answers - show all your work:

(2 points) your answer.

(1 points) justification for your choice.

Big O(1)

It does not matter where you insert the element in the array because it is unsorted.

1. Insert a new element into a sorted array.

Big O(logn) + Big O(n)= Big O(n)

Since the array is sorted, the element might have to be put at the beginning of the array. First you search, then you shift adding these two together gives you the answer.

1. Remove the minimum element in an unsorted array.

Big O(n)

Since the array is unsorted, the minimum element can be at the beginning of the array.

1. Remove the minimum element in a sorted array.

Big O(1)

Since the array is sorted, the minimum element will be found at the beginning or end of the array depending on how it is set up.

**TURN-IN CHECKLIST:**

1. **Answers to Section 1 (.doc/.txt/.pdf). Remember to include your name, the date, and the course number in comments near the beginning of your code/report.**
2. **Create a folder and name it 'FirstName\_LastName\_assignment\_3'. In the newly created folder copy and paste your files (.doc/.txt/.java files). Then compress the folder, and push it to iLearn.**