

Assignment 2

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

Section 1: Pen-and-paper Exercises

1. Analyze the following code and provide a "Big-O" estimate of its running time in terms of n . Explain your analysis.

```
int i = 1;
while (i <= n)
    some O(1) time statements;
    i = i*2;
end while
```

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

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

(2 points) your answer.

2. Analyze the following code and provide a "Big-O" estimate of its running time in terms of n . Explain your analysis.

```
for( int i = n; i > 0; i /= 2 ) {
    for( int j = 1; j < n; j *= 2 ) {
        for( int k = 0; k < n; k += 2 ) {
            ... // constant number of operations
        }
    }
}
```

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

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

(2 points) your answer.

3. Analyze the following code and provide a "Big-O" estimate of its running time in terms of n . Assume that $n = 2^m$. Explain your analysis.

```
for( int i = n; i > 0; i-- ) {
    for( int j = 1; j < n; j *= 2 ) {
        for( int k = 0; k < j; k++ ) {
            ... // constant number C of operations
        }
    }
}
```

Note:

$$2^0 + 2^1 + \dots + 2^k = 2^{k+1} - 1$$

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

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

(2 points) your answer.

4. Analyze the following code and provide a "Big-O" estimate of its running time in terms of n . Explain your analysis.

```
j = 1, i = 0;

while (i < n)
{
    i = i + j;
    j++;
}
```

Note: The loop variable 'i' is incremented by 1, 2, 3, 4, ... until i becomes greater than or equal to n.

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

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

(2 points) your answer.

5. Arrange the following functions in ascending order of growth rate (8 points):

n^4	$\sum_{i=1}^n 1$	$\log \log n$	2010	$\sum_{i=1}^n i$
2^n	\sqrt{n}	$\log n$	n^2	$n \log n$
n^n	$\sum_{i=1}^n \frac{1}{i}$	$n!$	e^n	n

You are NOT required to justify your ordering.

Note:

In this problem, you are asked to identify if $f_1(n) < f_2(n)$ for a "sufficiently large" input size n . However, for small values of n this is not always true.

6. Given a positive integer x , find square root of it. If x is not a perfect square, then return floor (round down).

Examples:

Input: $x = 4$

Output: 2

Input: $x = 11$

Output: 3

Outline an algorithm for finding square root of x . Expected in $O(\log n)$ time.

Full credit (10 points) will be awarded for an algorithm that is $O(\log n)$. Algorithms that are $O(n)$ or slower will be scored out of 5 points.

Note: You should NOT use existing functions like `math.sqrt()` to obtain the square root of x . Create your own function. Solutions that use existing functions will receive 0 points.

- (i) describe the idea behind your algorithm in English (2 points);**
- (ii) provide pseudocode (5 points);**
- (iii) analyze its running time (3 points).**

Section 2: Java Implementation

7. Implement problem 6 in Java (30 points).

Note:

Find a file called Problem6.java in assignment 2 folder.

Complete the method of `squareroot()`.

Test your method in the main method provided.

Programs that are $O(n)$ or slower will be scored out of 10 points.

Programs that use existing functions like `math.sqrt()` will receive 0 points.

Important: In all of the assignments of this course, when you are asked to implement an algorithm for a problem, your code will be evaluated based on:

5 points - Execution

Each file must run without error or warning on valid input described in the main method provided.

5 points - Within Code Documentation

Is the code documented for obvious understanding of the use, preconditions, and postconditions of each function?

20 points - Correctness

Is the algorithm implemented correctly? Does your method pass the test?

TURN-IN CHECKLIST:

- 1. Answers to Section 1 (.doc/.txt), and to Section 2 (all your source Code (.java files)). 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_2'. In the newly created folder copy and paste your files (.doc/.txt/.java files). Then compress the**

folder, and push it to iLearn.