**Name:\_Jacob\_Pacheco\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Assignment One**

Due in class Sept.7 ---All answers should be typed.

* **Rank the following big-O measures from most efficient (1) to least efficient (9):**

**Remember logan = ln(n)/ln(a)**

* O(n) : 4
* O(n3) : 8
* O(4n) : 9
* O(log4n) : 3
* O(log5n) : 2
* O(n2) : 7
* O(1) : 1
* O(nlog2n) : 5
* O(n2log2n) : 6

**2. Estimate the running time of the following algorithms using Big-O notation**

**a.** **for** (i = 1; i <= n; i++)

**for** (j = 1; j <= n; j++)

**for** (k = 1; k <= n; k++)

//five assignment statement

Answer: O(n^3)

**b** . **for** (i = 10; i <= n; i++)

**for** (j = 15; j <= n; j++)

**for** (k = 1; k <= n; k++)

//five assignments

Hint: It makes no difference whether we start the loop with 1 , 0, or 10

Answer: O(n^3)

**c.**

**int** i = 1;

**while** (i <= n)   
{

**int** j = 17;

**while** (j <= 100)

j++;

// Two assignments

i++;

}

Answer: O(n)

**d.**

**int** i = 1;

**while** (i <= n)   
{

**int** j = 1;

**while** (j <= n)   
 {

j = j \* 2;

//Two assignments

}

i++;

}

Answer: O(log\_2 n)

**3. Using big-O notation, estimate the running time of each of the following recursive functions. Here it is important to see how many times a recursive function will be called.**

**void** RecA (**int** n)   
{

//Some task requiring constant time

if (n > 0)

RecA(n-1);

}

Answer: O(n log n)  
  
**b.**

**void** RecB (**int** n)   
{

**int** i ;

**for** (i = 1; i <= n; i++)

//Some task requiring constant time

**if** (n > 1)

RecB(n-1);

}

Answer: O(n log n)

**c.**

**void** RecC (**int** n)   
{

**int** i;

**for** (i = 1; i <= n; i++)

//Some task requiring constant time

**if** (n > 1)

RecC(n/2)

}

Answer: O(n^2 log n)

**Problems from Page 147 in the textbook.**

**4.**

What is the Big-O

int sum = 0;

for( int counter = n; counter >0; counter = counter - 2)

sum = sum + counter;

Answer: O(n)

**5.** What is the Big-O:

int sum = 0;

for( int counter = n; counter >0; counter = counter \* 2)

sum = sum + counter;

Answer: SKIPPED

|  |
| --- |
| **6**. Using Big Oh notation, indicate the time requirement of each of the following tasks in the worst case. |
| a. Display all the integers in an array of integers. Answer: O(n) | |
| b. Display all the integers in a chain of linked nodes. Answer: O(n) | |
| c. Display the nth integer in an array of integers. Answer: O(1) | |
| d. Compute the sum of the first n even integers in an array of integers. Answer: O(n logn) | |