Homework 6

Due 10/28/2024

October 21, 2024

1. Use the formal definition of Big-Oh to prove the Largest Term property: if $f(n) = O(g(n))$, then $f(n) + g(n) = O(g(n))$.

2. Find the asymptotic Big-Theta growth rate for the functions $a(n)$, $b(n)$, $c(n)$, $d(n)$, and $e(n)$ based on the facts below (*5 answers*). You may assume all functions are nonnegative, and you may assume that if $f(n) = \Theta(g(n))$, then $f(n)^c = \Theta(g(n)^c)$ for any nonnegative real number c. You are **not** required to use the formal definition of Big-Theta. Show your work.

   - $a(n) = \Theta(d(n)e(n))$
   - $a(n)d(n)e(n) = \Theta(n^3)$
   - $b(n)^3 = \Omega(a(n)^2)$
   - $c(n) + d(n) = \Theta(b(n)^2)$
   - $d(n)^2 = \Theta(a(n)e(n))$
   - $e(n)^2 = \Omega(b(n))$

3. Analyze the worst-case time complexity of the LoopMystery algorithm below.

```
Input: n: nonnegative integer
Algorithm: LoopMystery
m = 0
for i = 1 to n do
    j = i
    while j < n do
        for k = j to n do
            m = m + 1
        end
        j = 2j
    end
end
return m
```

4. Find a recurrence that describes the worst-case complexity of the following recursive sorting algorithm. Show all work. You may assume that the floor function ($\lfloor\ \rfloor$) takes constant time.

```
Input: data: an array of integers
Input: n: the length of data
Output: a permutation of data such that
        data[1] ≤ data[2] ≤ ... ≤ data[n]
Algorithm: ThirdSort
if n = 1 then
    return data
else if n = 2 then
    if data[1] > data[2] then
        Swap data[1] and data[2]
    end
    return data
else
    third = ⌊n/3⌋
    Call ThirdSort on data[1..n-third]
    Call ThirdSort on data[third+1..n]
    Call ThirdSort on data[1..n-third]
    return data
end
```

5. Use the Master Theorem to find the worst-case complexity of ThirdSort and describe how ThirdSort compares to SelectionSort.

   You may assume that $f(n)$ is regular if relevant. Recall that $\log_a(b) = \frac{\ln(b)}{\ln(a)}$ (you may need a calculator for this one). Be sure to include the value of c and the case of the Master Theorem in your answer.

6. The *mode* of a dataset is the element that appears most frequently. Answer the following questions about developing a composite data structure to efficiently maintain and return the mode of the data it contains.

   (a) Describe the *data members* of a data structure that can return the mode of a dataset in $\Theta(1)$ time, add new elements in $O(\lg n)$ time, and remove elements in $O(\lg n)$ time (all worst case).

   Hint: the data members should be basic data structures we have covered in class. Note that the mode can change as elements are added to and removed from the dataset.

   (b) Give pseudocode for how to return the mode of your data structure. It should take $\Theta(1)$ time, worst case.

   (c) Give pseudocode for how to add an element to your data structure. It should take $O(\lg n)$ time, worst case.

(d) Give pseudocode for how to remove an element from your data structure. It should take $O(\lg n)$ time, worst case.