CS240 HW #2

Answers to the following exercises should be submitted through Canvas as a .pdf file. Don't forget to include your name and an honor code statement.

Hints for solving recurrences (these may or may not apply to the problems below):

- Two sums often show up in solving these recurrences. One is the geometric sum:

$$\sum_{j=0}^{n} r^j = \frac{1 - r^{n+1}}{1 - r}.$$

From which it follows that:

$$\sum_{j=0}^{n-1} r^j = \frac{1 - r^n}{1 - r}.$$

A second is the arithmetic sum:

$$\sum_{j=0}^{n} j = \frac{n(n+1)}{2}$$

From which it follows that:

$$\sum_{j=0}^{n-1} = \frac{(n-1)n}{2}.$$

- Remember that $a^{\log_2 b} = b^{\log_2 a}$. This helps, for instance, with simplifying statements like $4^{\log_2 n} = n^{\log_2 4} = n^2$.
- Wolfram alpha (www.wolframalpha.com) is able to solve recurrences like these automatically. Feel free to use this tool to check your answers. Your score will be based on showing the steps required to solve the recurrences using backward substitution.

1. Write the value of the following recurrences for $n = 0$ to 4. (3pts)

   (a) $T(0) = 1$
   $T(1) = 2$
   $T(n) = 2n + 2T(n-2)$

   (b) $T(0) = 1$
   $T(n) = 2T(n-1)$

(c) $T(0) = 1$
$T(1) = 1$
$T(n) = 3n^2 + n + T(n-2)$

2. Write the value of the following recurrences for $n = 1$, 2, 4 and 8. (3pts)

(a) $T(1) = 4$
$T(n) = 5T(n/2)$

(b) $T(1) = 4$
$T(n) = n + T(n/2)$

(c) $T(1) = 0$
$T(n) = n^2 + n + T(n/2)$

3. Find a closed form solution for each of the following recurrences (for b you may assume n is a power of 4). (4 pts)

(a) $T(0) = 2$
$T(n) = 1 + 4T(n-1)$

(b) $T(1) = 1$
$T(n) = n + 2T(n/4)$

4. Write recurrence relations that describe the number of times that `System.out.println` is called by each of the following recursive functions. (4pts)

```
public static void fun1(int n) {
    if (n == 0) {
        System.out.println("fun1 line");
    } else {
        for (int i = 0; i < 4; i++) {
            fun1(n / 2);
        }
    }
}

public static void fun2(int n) {
    if (n == 0) {
        System.out.println("fun2 line");
    } else {
        for (int i = 0; i < 3; i++) {
            System.out.println("fun2 line");
        }
        fun2(n - 1);
        fun2(n - 1);
    }
}
```

Another hint:

- Since we are counting println statements, you should be able to double check your solution by executing the code and counting the number of lines printed.

5. Use the method of backward substitution (or the tree method) to solve the recurrences from the previous exercise. Show your work. (4pts)

6. Design a recursive algorithm to solve the following problem. You are given an array A of n integers that is *monotonically decreasing*, meaning that for every index i from 0 to $n-2$, we have that $A[i] > A[i+1]$. Your task is to design a recursive algorithm to find the first negative number in A. Do the following:

   (a) (3 pts) Write detailed pseudo-code for your algorithm with a comment for every line of code explaining what it does.

   (b) (2 pts) Write a recurrence for the running time of your algorithm.

   (c) (2 pts) Find the closed form of the recurrence. Use it to evaluate the efficiency of your algorithm. Your algorithm should run in $\Theta(\log n)$ time. If yours is slower, you need to design a better solution.