

## CS240 HW #4

Answers to the following exercises should be prepared in a text editor and submitted through Canvas as a .pdf file. Equations should be properly formatted using equation editing software. Don't forget to include your name and an honor code statement.

1. Find a closed form solution to each of the following recurrences. Remember to show all of your work. You may choose either the substitution method or the tree method. You should expand out to at least 3 expansions, give the general form in terms of  $i$  expansions, determine how many expansions or tree levels you have to perform to get to a base case, and give the closed form solution for the recurrence. (4 pts each)

Note: for a full solution, we expect that sums will be evaluated (i.e. you should turn things like  $\sum_{i=0}^n 2^i$  into  $2^{n+1} - 1$ ).

- $T(1) = 2$   
 $T(n) = T(n/2) + 2n$   
(You may assume  $n$  is a power of 2.)
- $T(0) = 1$   
 $T(n) = 3T(n-1) + 1$
- $T(1) = 1$   
 $T(n) = 3T(n/4) + 1$   
(You may assume  $n$  is a power of 4.)
- $T(1) = 1$   
 $T(n) = 6T(n/3)$   
(You may assume  $n$  is a power of 3.)

2. Write recurrence relations that describe the number of times that `printf` is called by each of the following recursive functions. (2pts each)

```
void fun1(int n):
    if (n == 0) {
        printf("%d\n", n);
    } else {
        for (int i = 0; i < 3; i++) {
            fun1(n - 1);
        }
        printf("%d\n", n);
    }
}

void fun2(int n) {
    if (n == 1) {
        printf("%d\n", n);
    } else {
        int i = 1;
        while (i <= 6) {
            fun2(n/3);
            i++;
        }
    }
}
```