CSE 332: Data Structures and Parallelism

QuickCheck: Recurrences (due Thursday, January 19)

Name:

Master Theorem

Consider a recurrence of the form

$$T(n) = \begin{cases} d & \text{if } n = 1 \\ aT\left(\frac{n}{b}\right) + n^c & \text{otherwise} \end{cases}$$

Then,

- If $\log_b(a) < c$ then $T \in \Theta(n^c)$
- If $\log_b(a) = c$ then $T \in \Theta(n^c \lg(n))$
- If $\log_b(a) > c$ then $T \in \Theta(n^{\log_b(a)})$

0. Sum Sum Sum

Consider the following code:

```
f(n) {
   if (n == 0) {
      return 0
   }
   int result = 0
   for (int i = 0; i < n; i++) {
      result += i * i + n
   }
   return f(n/3) + 2 * result + 3 * f(n/3)
}
```

(a) Find a recurrence $T(n)$ modeling the worst-case time complexity of $f(n)$.

(b) Find a Big-Oh bound for your recurrence.