CS240E S26	Tutorial 1	May 15

- Asymptotic notation review
- Two approaches to $n \in \omega\left(2^{\sqrt{\lg n}}\right)$
- Runtime analysis examples
- Dealing with recursive formulas

Q1. Show that $n \in \omega\left(2^{\sqrt{\lg n}}\right)$ using the definition.

Q2. Give a tight bound on the runtime of the following algorithm as a function of n.

```
k = 1
for(i = 1; i <= n; ++i):
    j = 0
    while j <= i * n:
        j += k
    k *= 2
```

Q3. There are two classic algorithms for matrix multiplication, the so-called "divide-and-conquer" and "Strassen" algorithms, whose running times are described by the following recurrences:

(1) $T(n) = 8T(n/2) + n^2$

(2) $T(n) = 7T(n/2) + n^2$

respectively. Assuming any appropriate base cases, show that these recurrences resolve to $\Theta(n^3)$ and $\Theta(n^{\log 7})$ respectively.

Q4. Assuming any appropriate base case, give asymptotic upper and lower bounds (make them as tight as you can) for $T(n)$ if,

(a) $T(n) = 3T(n/3) + n/\log_3 n$

(b) $T(n) = T(n-1) + 1/n$

(c) $T(n) = \sqrt{n}T(\sqrt{n}) + n$