ECS 122A
Algorithm Design and Analysis

Instructor: Qirun Zhang

Course slides (partially) adopted from the notes by David Luebke.

Agenda

- Substitution method
- Master method

Course updates

- Feedback
- About homework
  - Will be posted today
  - Submit 5 separate solutions on gradescope (i.e., one for each problem)
- Prerequisite petition
  - Reply my email

The Master Theorem

- if T(n) = aT(n/b) + f(n) then

$$T(n) = \begin{cases} \Theta\left(n^{\log_b a}\right) & f(n) = O\left(n^{\log_b a - \varepsilon}\right) \\ \Theta\left(n^{\log_b a} \log n\right) & f(n) = \Theta\left(n^{\log_b a}\right) \\ \Theta\left(f(n)\right) & f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right) \text{ AND} \\ & af(n/b) < cf(n) \text{ for large } n \end{cases} \quad \begin{matrix} \varepsilon > 0 \\ c < 1 \end{matrix}$$

Using The Master Method

- $T(n) = 9T(n/3) + n$
  - $a=9$, $b=3$, $f(n) = n$
  - $n^{\log_b a} = n^{\log_3 9} = \Theta(n^2)$
  - Since $f(n) = O(n^{\log_3 9 - \varepsilon})$, where $\varepsilon=1$, case 1 applies:
  - Thus the solution is $T(n) = \Theta(n^2)$

$$T(n) = \Theta\left(n^{\log_b a}\right) \text{ when } \quad f(n) = O\left(n^{\log_b a - \varepsilon}\right)$$

The End