CS 313
Instructor: Kangmei Yang
Recurrence Relations

- In the analysis of algorithm, some are not simple statement in sequential order.
- Recursive algorithm like following:
  ```java
  void method(int n){
    if (n < 1) return;
    method(n/b);
    method(n/b);
    method(n/b);
    ...repeat for a total of a times...
    method(n/b);
  }
  ```
- Divide problem into a number of subproblems
- Each subproblem ha a size of n/b
- Such algorithms can be represented as a recurrence relation
Recurrence Relations

- Recurrence relation can be expressed as following,

\[
\begin{cases}
    c & \text{if } n < 1 \\
    a T(n / b) + f(n) & \text{if } n \geq 1
\end{cases}
\]

- Master Theorem are used to analyze the time-complexity for such form where \( a \geq 1, b > 1 \)
- \( n \) is the size of the problem
- \( a \) is the number of subproblems in the recursion
- \( n/b \) is the size of each subproblem
- \( f(n) \) is the cost of the work done outside of the recursive call for each iterative step
Master Theorem

- The time complexity for recurrence relation as following,

\[
\begin{align*}
\Theta(n^{\log_b a}) & \quad \text{if } c < \log_b a \\
\Theta(n^c \log n) & \quad \text{if } c = \log_b a \\
\Theta(n^c) & \quad \text{if } c > \log_b a
\end{align*}
\]