# §1 Lecture 01-14

## §1.1 Recurrences

Mergesort: $T_n = 2T_{n/2} + n$. Roughly because emitting the floor function.
Binary Search: $T_n = 2T_{n/2} + 1$. Roughly because emitting the floor function.
Chip Testing: $T_n \leq T_{n/2} + \frac{3n}{2}$.
Karatsuba Multiplication: $T_n = 3T_{n/2} + n$.
Matrix Multiplication Strassen: $T_n = 7T_{n/2} + n^2$
Halfspace Counting: $T_n = 3T_{n/4} + 1$

## §1.2 Solving Recurrences

Methods include:

1. Exact Methods
    a) Substitution. Summing a series.
    b) Induction. Assume $T_n = f(n)$.
2. Order of magnitude
    a) Master theorem: Gives $O(), \Theta(), \Sigma()$ result. Order of magnitude.
    b) Recursion tree: to get insight.

**Example 1.1** (Recursion Tree)

$$T_n = 3T_{\frac{n}{4}} + n$$
$$T_1 = 1 \text{ or } 0, T_0 = 0$$
$$k \text{ (height) } = \log_4(n)$$
$$\frac{n}{4^k} = 1$$
$$n, \frac{3}{4}n, (\frac{3}{4})^2 n, (\frac{3}{4})^3 n, \ldots Sum = n(1 + \frac{3}{4} + (\frac{3}{4})^2 + \cdots + (\frac{3}{4})^k)$$
$$\approx= n(1 + \frac{3}{4} + (\frac{3}{4})^2 + \cdots) = \frac{1}{1 - \frac{3}{4}} = 4n \leq 4n \leq 4n + o(n)$$
$$\sum_{i=0}^{\infty} x^i = \frac{1}{1-x}$$
$$\Rightarrow T_n = \Theta(n)$$

Most of the time is spent at the top level in this problem

Changing the problem

$$T_n = 4T_{\frac{n}{4}} + n$$
$$n, n, n, n, n, \ldots$$
$$\Rightarrow T_n = \Theta(n \log n)$$

The same amount of time is spent at each level in this problem

Changing one more time

$$T_n = 5T_{\frac{n}{4}} + n$$
$$n, (\frac{5}{4})n, (\frac{5}{4})^2 n, \ldots$$
$$Sum = n(1 + \frac{5}{4} + (\frac{5}{4})^2 + \cdots + (\frac{5}{4})^k)$$
$$Sum = n(\frac{5}{4})^k(1 + \frac{4}{5} + (\frac{4}{5})^2 + \cdots + (\frac{4}{5})^k) \approx \frac{1}{(1 - \frac{4}{5})} = 5$$
$$\approx n(\frac{5}{4})^k * 5 = 5 * 5^k = 5 * 5^{\log_4(n)} = 5n^{\log_4(5)}$$

More time is spent at the bottom level in this problem

Note that the coefficient in front of n wouldn't affect the magnitude.

## §1.3 Master Theorem

$$T_n = aT_{n/b} + f(n) \quad (T_n = constant, n \leq \square)$$

Largest of $f(n), n^{\log_b(a)}$

1. $$\frac{n^{\log_b(a)}}{f(n)} \geq n^\epsilon \text{ for some } \epsilon > 0 : T_n = \Theta(n^{\log_b(a)})$$

2. $$\frac{f(n)}{n^{\log_b(a)}} \geq n^\epsilon .... \quad T_n = \Theta(f(n))$$

3. $$f(n) = \Theta(n^{\log_b(a)}) : T_n = \Theta(f(n) \times \log(n))$$

Technical Condition

$$\limsup_{n \to \infty} \frac{af(\frac{n}{b})}{f(n)} < 1$$

If $T_n \leq aT_{n/b} + f(n) \quad (T_n = constant, n \leq \square)$, then $O()$ instead of $\Theta()$. If $\geq$, $\Sigma()$.
Other cases (exercises to think about):

$$T_n = T_{n/2} + \log n$$
$$T_n = T_{n/2} + T_{n/3} + n$$

## §1.4 Strassen's Matrix Multiplication

For any $n$, there exists a power of $2 \leq 2n$. So assume that $n$ is a power of 2 without loss of generality. Strassen gets rid of a multiplication.

$$T_n = 8T_{\frac{n}{2}} + n^2 = \Theta(n^3)$$
Became
$$T_n = 7T_{\frac{n}{2}} + n^2 = \Theta(n^{2.8})$$

Best known to date: $\Theta(n^{2.374})$. Goal: $\Theta(n^2 \log n)$.

## §1.5 Halfspace Counting

Given: $x_1, \ldots, x_n \in \mathbb{R}^2$. Users query: How many points on one side of $H$ (hyperplane).
Fact: There exists a partition of $x_1, \ldots, x_n$ into 4 equal sets, using only 2 lines.

**Theorem 1.2** (Pancake Theorem)
Take a shape. There is always a cut that cuts the pancake into two equal sets.
Also works with overlapping pancakes. Can cut both in have with a single line.