(not class Monday) - wednesday we start anchor algorithms.

class today

- pdf of lectures should be up after

- hwo due wednesday

Announcement

CS 314 - Lecture 3
Today - Recursion

If induction starts at base case & builds up, then recursion is dual idea:

**IS** - Start with n things

**IH** - Reduce to (a) smaller subproblem(s)

**BC** - Eventually reach smallest case aka base case.
Solving recurrences

Method 1: guess & check

Ex: Towers of Hanoi

\[ T(n) = T(n-1) + 1 + T(n-1) \]

\[ \rightarrow 2^n - 1 \text{ moves} \]
\[
T(1) = 3 \quad T(2) = 3 \quad T(3) = 2(3) + 1 = 7
\]

\[
T(4) = 2T(3) + 1 = 2(7) + 1 = 15
\]

\[
T(5) = 2T(4) + 1 = 2(15) + 1 = 31
\]

\[
T(6) = 2T(5) + 1 = 2(31) + 1 = 63
\]

\[
T(7) = 2T(6) + 1 = 2(63) + 1 = 127
\]

Fill in some values.

Write as recurrence:

\[
T(n) = 2T(n-1) + 1
\]

How many moves?
\[ 1 - c^n = \left(1 + e^{-u} \right) \left(1 + (1 - e^{-u}) \cdot (1 + \sum_{k=1}^{n-1} (1 - e^{-u})^k) \right) \]

Initialize:
\[ T(0) = e \]

Use recurrence:
\[ T(n) = e \cdot T(n-1) \]

Prove:\n\[ T(n) = e^n - 1 \]

Proof (by induction):\n\[ T(n) = e^n - 1 \]
10 \rightarrow 0 \rightarrow \frac{10}{\log 2}

\text{Answer:}

T(n) = \begin{cases} 
\log n & \text{if } n = 1 \\
T(\frac{n}{2}) + T(\frac{n}{2}) + \Theta(n) & \text{otherwise}
\end{cases}

\text{Guess + check - merge-sort}
\[ T(n) = \frac{3n}{n} + \frac{1}{n} \]

\[ T(\frac{n}{2}) = \frac{2 \sqrt{n} - \frac{\sqrt{n}}{4}}{n} + \frac{3}{n} \]

\[ T(\frac{n}{2}) = 4 \left( \frac{n}{4} \right) + \frac{3}{n} \]

When \( n = \text{power of } 2 \), we have \( \log n \text{ base } 2 \) = \( \frac{\log n}{\log 2} \).

Assume \( n = \text{power of } 2 \) in exponent, \( \log n \text{ base } 2 \).

Can't skip the base-0 as used induction.
\( T(n) = 2T\left(\frac{n}{2}\right) + n \)
\[ t(n) = a + (\frac{n}{n_0} + f(n)) \]

where:

- \( a \) is a constant, \( f(n) \) is a function.

\[ t(n) = a + (\frac{n}{n_0} + f(n)) \]

Divide & Conquer Recurrence Often

Recursion Trees
Picture:

\[ L(k) = 0.7 \left( \frac{a}{b} \right) + f(k) \]
Gives a Sum? (Is it like those...)
\( E \) : MergeSort \( T(n) = 2T(n/2) + n \)

\( a = 6 \Rightarrow 2 \)
\[
\log_{\frac{3}{4}} n = \log_{\frac{3}{4}} \left( \frac{3}{4} \right)^n
\]

Depth d

Randomized Selection: \( T(n) = T\left(\frac{n}{3}\right) + n \)
\[ T(n) = \Theta(f(n) \log n) \]

3) \[ \text{If } a_f(n) = f(n), \text{ then } T(n) = \Theta(n) \]

2) \[ \text{If } a_f(n) = K \cdot f(n) \text{ and } K > 1, \text{ then } T(n) = \Theta(K \cdot f(n)) \]

1) \[ \text{If } a_f(n) = f(n) \text{ and } f(n) = \Omega(\log n), \text{ then } T(n) = \Theta(f(n) + \log n) \]

Master Theorem: Suppose \[ T(n) = a \cdot T(n) + f(n) \]
Geometric Series

Why?
Ex: $T(n) = \sum T(n/2) + n$

For those:

Can use recursion trees even

Sahsy Mexta Truu!

Caution: Not all recursive