CS 305
Design and Analysis of Algorithms

09 / 23 / 2021

Instructor: Michael Eckmann

Today's Topics

- Questions / Comments?
- Master Method of solving recurrences
  - Prove case 3
  - Do some examples
- Start Quicksort

Master Method for Recurrences

- The Master Method can be used to solve recurrences of the form: $T(n) = a * T(n/b) + f(n)$
- There are 3 cases to consider each based on a relationship of time spent at leaves vs. at the root.

Master Method for Recurrences

- Recurrences of the form: $T(n) = a * T(n/b) + f(n)$

- Case 1: if time spent at root is "at most little bit smaller" than time spent at all the leaves, then the running time of algorithm is dominated by the time spent at the leaves
- Case 2: if time spent at root is big theta (same as) of time spent at all the leaves then it is lgn * time spent at leaves
- Case 3: if time spent at root is "at least little bit larger" than time spent at all the leaves, then the running time of algorithm is dominated by the time spent at the root

Master Method for Recurrences

- We have already proved cases 1 and 2
- Let's prove case 3 of master theorem which will tell us the running time when time spent at root is "at least little bit larger" than time spent at all the leaves (with an added constraint)

Quicksort / Partition examples

- Problem with MergeSort is space is BigTheta(n)
- Quicksort is a divide and conquer algorithm that works in-place (that is, with space BigTheta(1))
- Example on the board of Partition function