The master theorem

Goldsmiths Computing

November 2, 2018

Motivation

- (asymptotically) solves recurrence relationships
- including:
  - straightforward ones (from first year)
  - harder ones (from this course)
- you need to:
  - know the result
  - be able to apply the result
- (you don't need to know the proof)

Examples

Binary search

$$T(N) = T\left(\frac{N}{2}\right) + O(1)$$

Binary tree traversal

$$T(N) = 2T\left(\frac{N}{2}\right) + O(1)$$

Merge sort

$$T(N) = 2T\left(\frac{N}{2}\right) + O(N)$$

Recursion trees

$$T(N) = aT\left(\frac{N}{b}\right) + f(N)$$

$$T(N)$$

Recursion trees

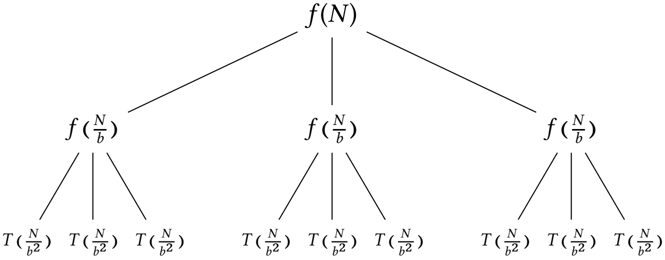
$$T(N) = aT\left(\frac{N}{b}\right) + f(N)$$

$$f(N)$$

$$T\left(\frac{N}{b}\right) \qquad T\left(\frac{N}{b}\right) \qquad T\left(\frac{N}{b}\right)$$

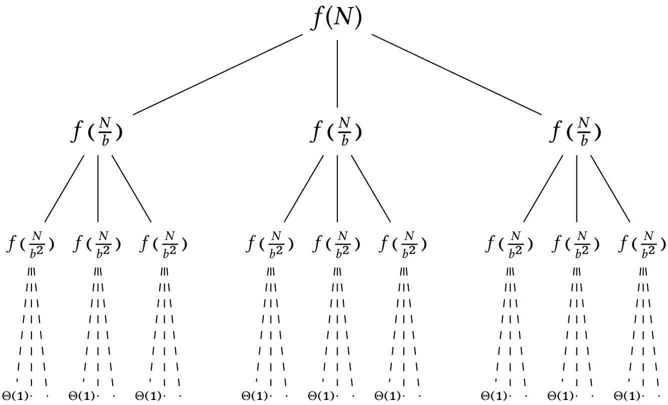
Recursion trees

$$T(N) = aT\left(\frac{N}{b}\right) + f(N)$$



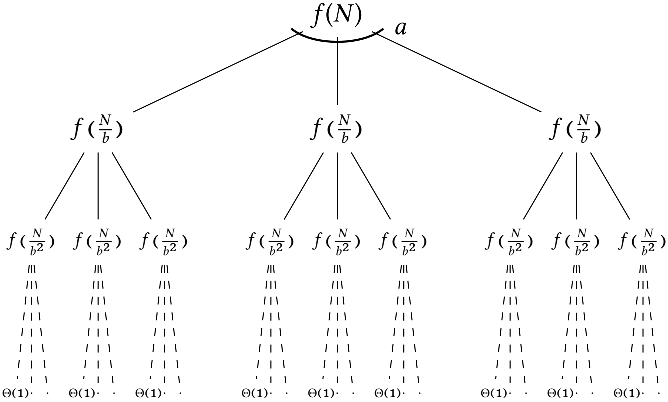
Recursion trees

$$T(N) = aT\left(\frac{N}{b}\right) + f(N)$$



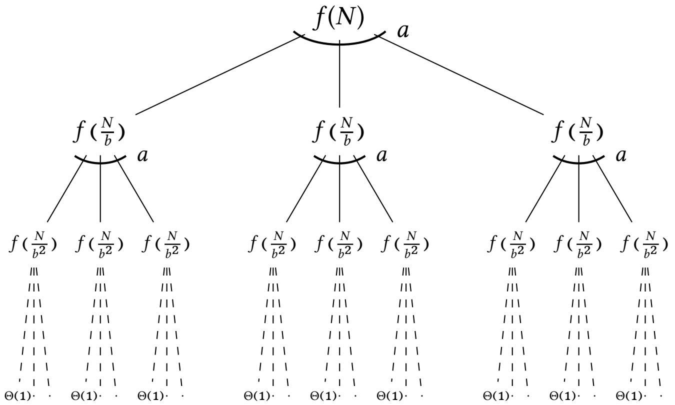
Recursion trees

$$T(N) = aT\left(\frac{N}{b}\right) + f(N)$$



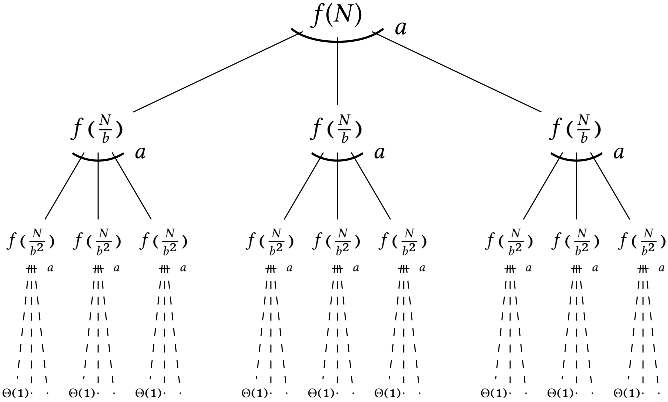
Recursion trees

$$T(N) = aT\left(\frac{N}{b}\right) + f(N)$$



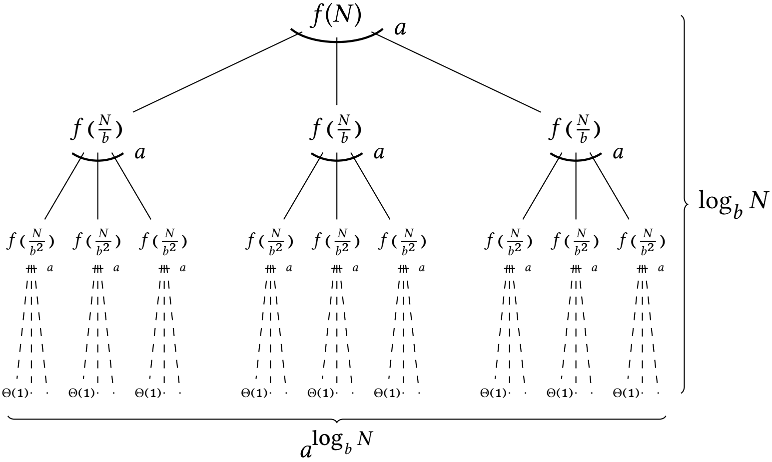
Recursion trees

$$T(N) = aT\left(\frac{N}{b}\right) + f(N)$$



Recursion trees

$$T(N) = aT\left(\frac{N}{b}\right) + f(N)$$



Theorem statement

$$T(N) = aT\left(\frac{N}{b}\right) + f(N)$$

Three cases:

1. $f(N) \in O\left(N^c\right)$ where $c < \log_b a$
2. $f(N) \in \Theta\left(N^c \log^k N\right)$ where $c = \log_b a$
3. $f(N) \in \Omega\left(N^c\right)$ where $c > \log_b a$

Theorem statement, case 1

$$T(N) = aT\left(\frac{N}{b}\right) + f(N)$$

$$f(N) \in O\left(N^c\right) \text{ where } c < \log_b a$$

Result

$$T(N) \in \Theta\left(N^{\log_b a}\right)$$

Example (binary tree traversal)

$$T(N) = 2T\left(\frac{N}{2}\right) + 1$$

$$a = 2, b = 2, \log_b a = 1; f(N) = 1 \in O(N^0)$$

so

$$T(N) \in \Theta(N^1) = \Theta(N)$$

Theorem statement, case 2

$$T(N) = aT\left(\frac{N}{b}\right) + f(N)$$

$$f(N) \in \Theta\left(N^c \log^k N\right) \text{ where } c = \log_b a$$

Result

$$T(N) \in \Theta\left(N^{\log_b a} \log^{k+1} N\right)$$

Example (binary search)

$$T(N) = T\left(\frac{N}{2}\right) + 1$$

$$a = 1, b = 2, \log_b a = 0; f(N) = 1 \in \Theta(N^0)$$

so

$$T(N) \in \Theta(N^0 \log^1 N) = \Theta(\log N)$$

Theorem statement, case 2

$$T(N) = aT\left(\frac{N}{b}\right) + f(N)$$

$$f(N) \in \Theta\left(N^c \log^k N\right) \text{ where } c = \log_b a$$

Result

$$T(N) \in \Theta\left(N^{\log_b a} \log^{k+1} N\right)$$

Example (merge sort)

$$T(N) = 2T\left(\frac{N}{2}\right) + N$$

$$a = 2, b = 2, \log_b a = 1; f(N) = N \in \Theta(N^1)$$

so

$$T(N) \in \Theta(N^1 \log^1 N) = \Theta(N \log N)$$

Theorem statement, case 3

$$T(N) = aT\left(\frac{N}{b}\right) + f(N)$$

$$f(N) \in \Omega\left(N^c\right) \text{ where } c > \log_b a$$

Result

$$T(N) \in \Theta\left(f(N)\right)$$

Example (quickselect)

$$T(N) = T\left(\frac{N}{2}\right) + N$$

$$a = 1, b = 2, \log_b a = 0; f(N) = N \in O(N^1)$$

so

$$T(N) \in \Theta(f(N)) = \Theta(N)$$

Proof

See CLRS for details:

- polynomially smaller/bigger;
- handling floor $\lfloor x \rfloor$ and ceiling $\lceil x \rceil$;
- regularity condition for case 3;

See also:

- Mohamad Akra and Louay Bazzi, *On the solution of linear recurrence equations*, Computational Optimization and Applications 10(2):195–210, 1998

Work

1. draw out the recursion trees for each of the examples given in the lecture (binary tree traversal, binary search, merge sort, quickselect) and convince yourself by adding up the contributions at each of the nodes that the results given in the lecture are correct.
2. Reading
   - CLRS, sections 4.4 and 4.5
3. Questions from CLRS

   Exercises 2.3-3, 2.3-4, 4.5-1, 4.5-2

   4-1 Recurrence examples

   4-2 Parameter passing costs

   4-3 More recurrence examples
4. do the quiz on Recurrence Relations on the VLE
   - open from 19th November
   - as many attempts as you like (but: 4 hours between attempts)