Design and Analysis of Algorithms

Lecture-6

Dharmendra Kumar (Associate Professor)
Department of Computer Science and Engineering
United College of Engineering and Research,
Prayagraj

Recurrence tree method

- In a ***recursion tree***, each node represents the cost of a single sub-problem somewhere in the set of recursive function invocations. We sum the costs within each level of the tree to obtain a set of per-level costs, and then we sum all the per-level costs to determine the total cost of all levels of the recursion.

- A recursion tree is best used to generate a good guess, which you can then verify by the substitution method.

Recurrence tree method

Example: Solve the following recurrence equation

$$T(n) = 3T(\lfloor n/4 \rfloor) + \theta(n^2)$$

Solution:

Recurrence tree method

Example: Solve the following recurrence relation using recurrence tree method

$$T(n) = T(n/3) + T(2n/3) + \theta(n)$$

Solution:

Recurrence tree method

Exercise

(1) Draw the recursion tree for $T(n) = 4\,T(\lfloor n/2 \rfloor) + cn$, where c is a constant and provide a tight asymptotic bound on its solution. Verify your bound by the substitution method.

(2) Use a recursion tree to give an asymptotically tight solution to the recurrence $T(n) = T(n-a) + T(a) + cn$, where $a \geq 1$ and $c > 0$ are constants.

(3) Use a recursion tree to give an asymptotically tight solution to the recurrence $T(n) = T(\alpha n) + T((1-\alpha)n) + cn$, where α is a constant in the range $0 < \alpha < 1$ and $c > 0$ is also a constant.