CS207 Design and Analysis of Algorithms

Sajith Gopalan

Indian Institute of Technology Guwahati

sajith@iitg.ac.in

January 18, 2022

Divide and Conquer

Divide and Conquer

- solve a problem recursively
  - Divide the problem into a number of smaller instances of the same problem
  - For each of these subproblems, if its size is sufficiently large, then Conquer it recursively, else Conquer it directly
  - Combine the solutions to the subproblems into the solution for the original problem

Lemma 1

Lemma

For positive constants a and b and integer valued functions $f(n)$ and $T(n)$ on exact powers of b, if

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ aT(n/b) + f(n) & \text{if } n = b^i \text{ for integer } i > 0 \end{cases}$$

then

$$T(n) = \Theta(n^{\log_b a}) + \sum_{k=0}^{log_b n - 1} a^k f(n/b^k)$$

Proof

$$T(n) = aT(n/b) + f(n)$$

$$= a[aT(n/b^2) + f(n/b)] + f(n) = a^2T(n/b^2) + af(n/b) + f(n)$$

$$= a^3T(n/b^3) + a^2f(n/b^2) + af(n/b) + f(n)$$

$$= a^jT(n/b^j) + a^{j-1}f(n/b^{j-1}) + \ldots + f(n)$$

$$= a^{\log_b n}T(1) + \sum_{k=0}^{log_b n-1} a^k f(n/b^k) \text{ (When } j = \log_b n)$$

$$= \Theta(n^{\log_b a}) + \sum_{k=0}^{log_b n-1} a^k f(n/b^k)$$

Lemma 2

Lemma

For positive constants a and b and integer valued functions $f(n)$ and $T(n)$ on exact powers of b, if

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ aT(n/b) + f(n) & \text{if } n = b^i \text{ for integer } i > 0 \end{cases}$$

and $f(n) = O(n^{\log_b a - \epsilon})$, for a constant $\epsilon \in (0, 1)$, then

$$T(n) = \Theta(n^{\log_b a})$$

Proof

$$\sum_{k=0}^{log_b n-1} a^k f(n/b^k) = O\left(\sum_{k=0}^{log_b n-1} a^k (n/b^k)^{\log_b a-\epsilon}\right) =$$

$$O\left(n^{\log_b a-\epsilon} \sum_{k=0}^{log_b n-1} a^k / b^{k \log_b a - k\epsilon}\right) = O\left(n^{\log_b a-\epsilon} \sum_{k=0}^{log_b n-1} b^{k\epsilon}\right) =$$

$$O\left(n^{\log_b a-\epsilon}.\frac{b^{\epsilon \log_b n} - 1}{b^{\epsilon} - 1}\right) = O\left(n^{\log_b a-\epsilon}.\frac{n^{\epsilon} - 1}{b^{\epsilon} - 1}\right) = O\left(n^{\log_b a}\right)$$

Lemma 3

Lemma

For positive constants a and b and integer valued functions $f(n)$ and $T(n)$ on exact powers of b, if

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ aT(n/b) + f(n) & \text{if } n = b^i \text{ for integer } i > 0 \end{cases}$$

and $f(n) = \Theta(n^{\log_b a})$, then

$$T(n) = \Theta(n^{\log_b a} \log_2 n)$$

Proof

$$\sum_{k=0}^{log_b n-1} a^k f(n/b^k) = \Theta\left(\sum_{k=0}^{log_b n-1} a^k (n/b^k)^{\log_b a}\right) =$$

$$\Theta\left(n^{\log_b a} \sum_{k=0}^{log_b n-1} a^k / b^{k \log_b a}\right) = \Theta\left(n^{\log_b a} \sum_{k=0}^{log_b n-1} 1\right) =$$

$$\Theta\left(n^{\log_b a} . \log_b n\right) = \Theta\left(n^{\log_b a} . \log_2 n\right)$$

Lemma 4

Lemma

For positive constants a and b and integer valued functions $f(n)$ and $T(n)$ on exact powers of b, if

$$T(n) = \begin{cases} \Theta(1) & \textit{if } n = 1 \\ aT(n/b) + f(n) & \textit{if } n = b^i \textit{ for integer } i > 0 \end{cases}$$

$f(n) = \Omega(n^{\log_b a + \epsilon})$, *for a constant* $\epsilon \in (0, 1)$, *and*
there exists $c < 1$ *and* $n_0 > 0$ *such that for all* $n \geq n_0$,
$af(n/b) \leq cf(n)$ *then*

$$T(n) = \Theta(f(n))$$

Proof

$$\exists c < 1, \exists n_0 > 0, \forall n \geq n_0, [af(n/b) \leq cf(n)]$$

$$\implies \exists c < 1, \exists n_0 > 0, \forall n \geq n_0, [f(n/b) \leq (c/a)f(n)]$$

$$\implies \exists c < 1, \exists n_0 > 0, \forall n \geq bn_0, [f(n/b^2) \leq (c/a)^2 f(n)]$$

$$\implies \exists c < 1, \exists n_0 > 0, \forall n \geq b^{j-1}n_0, [f(n/b^j) \leq (c/a)^j f(n)]$$

That is, $f(.)$ at a sufficiently large divisor n/b^j of n (which is a power of b), is upper bounded by $(c/a)^j f(n)$
For smaller divisors of n, $f(.)$ is upper bounded by a constant.

Proof

$$\sum_{k=0}^{log_b n-1} a^k f(n/b^k) \le \sum_{k=0}^{log_b n-1} c^k f(n) + O(1) \le$$

$$f(n)\sum_{k=0}^{\infty} c^k + O(1) = f(n)\frac{1}{1-c} + O(1) = \Theta(f(n))$$