MIT World Peace University

Analysis of Algorithms

Unit 1

Naman Soni Roll No. 10

Contents

1 Divide and Conquer

1.1 Control Abstraction

```
DANDC (P)
{
    if SMALL (P) then return S (p);
    else
    {
        divide p into smaller instances p1, p2,...Pk, k>=1;
        apply DANDC to each of these sub problems;
        return (COMBINE (DANDC (p1), DANDC (P2),...,DANDC (pk)));
    }
}
```

1.2 Time Complexity of the general algorithm

- A recurrence is an equation or inequality that describes a function in terms of its value on smaller inputs.
- Special techniques are required to analyze the space and time required.
- $T(n) = \frac{aT(\frac{n}{b}+1)(n)+c(n)}{O(1)}$
- Time Complexity (recurrence relation): (
  - where D(n): time for splitting
  - C(n): time for conquer
  - c: a constant

)

1.3 Methods for Solving recurrences

1. Substitution method: This method involves guessing a solution and then proving that it is correct.
2. Recurrence tree method: This method involves constructing a tree diagram that represents the recursive calls and their relationship to each other.
3. Master theorem: This is a general theorem that provides a method for solving recurrences of a specific form.

1.4 Math you need to Review

Properties of Logarithms:

- $\log_b(xy) = \log_b(x) + \log_b(y)$
- $\log_b(\frac{x}{y}) = \log_b(x) - \log_b(y)$
- $\log_b xa = a \log_b x$
- $\log_b a = \frac{\log_x a}{\log_b b}$

Properties of exponentials:

- $a^{(b+c)} = a^b a^c$
- $a^{bc} = (a^b)^c$

- $\frac{a^b}{a^c} = a^{(b-c)}$
- $b = a^{\log_a b}$
- $b^c = a^{c^* \log_a b}$