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:

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

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

2 Divide-and-Conquer Examples

- Sorting: mergesort and Quick Sort
- Binary tree traversals
- Binary search
- Multiplication of Large integers
- Matrix Multiplicatoion: Strassen's algorithm
- Closet-pair and convex-hull algorithms

3 Proof Techniques

- Proof is a kind of demonstration to convince that the given mathematical statement is true.
- The statement which is to be proved is called theorem. Once a particulare theorem is proved then it can be used to prove further statements.
- The theorem is also called Lemma.
- The proof can be a deductive proof or inductive proof.
- The deductive proof consist of sequence of statements given with logical reasoning.
- The inductive proof is a recursive kind of proof which consists of sequence or parameterized statements that use the statement itself or the statement with lower values of its parameters/.

3.1 Proof by contradiction

- In this type of proof, for the statement of this form is A and the B.
- Prove by contradiction. There exist two irrational numbers x and y such that x^y is rational.
  - An irrational number is any number taht cannot be expressed as a/b where a and b are integers and value b is non zero. To prove that x^y is rational when x and y are rational numbers.

3.2 Proof by Mathematical Induction

- Prove $1 + 2 + 3... + n = n(n + 1)/2$
  - 1) Basis of incution
  - Assume, n = 1 then
  - LHS = n = 1
  - RHS =$n(n + 1)/2 = 1(1 + 1)/2 = 2/2 = 1$
  - 2) Induction hypothesis
  - Now we will assume n = K and will obtain the result for it. The equation then becomes,
  - $1 + 2 + 3 + ... + K = K(K = 1)/2$

- 3) Inductive step
- Now we assume that equation is true for n = K and we willl then check if it is also true for n = K + 1 or not.
- Consider the equqtion assuming n = K + 1
- LHS = 1 + 2 + 3 + ... + K + K + 1
- $= K(K+1)/2 + K + 1$
- $= K(K+1) + 2(K+1)/2$
- $= (K+1)(K+2)/2$

3.3 Direct Proof

- In direct Proof, the intended proof can be proved by basic principle or axiom.
- Example:- Prove that the negative of any even integer is even.
- Solution: to prove this, let n be any positive even number. Hence we can write n as
- $n = 2m$ where m can be any number
- if we multiply both side by -1, we get
- $-n = -2m$
- $-n = 2(-m)$
- Multiplying any number by 2 makes it an even number.
- Hence, -n is even.
- Thus proves that the negative of any even integer is even.

3.4 Proof by Counter-Example

- This is a technique of proof in which a-¿b is true if a-¿ b
- If negative statement of given statemetn is true then the given statement becomes automatically true.
- Example: prove contraposition that $x + 8$ is odd
- Solution:
  - Step 1: we assume that x is not odd
  - Step 2: that means x is even. By definition of even numbers 2 * any number = even number.
  - $x = 2 * m$ where m can be any number
  - we can write $x + 8$ as $2 * m + 8 = 2(m + 4) = evennumber$
  - thus, $x + 8$ is even

4 Brute Force

- Burte force is a straightforward approach to solve a problem based on the problem's statement and definitions of the concepts involved.
- It is condidered as one of the easiest approach to apply and is useful for solving small-size instances of a problem.

4.1 Brute Force Search and Sort

- Sequential search in an unordered array and simple sorts- selection sort, bubble sort are brute force algorithm.
- Sequential search: the algorithm simply compared successive elements of a given list with a given search key until either a match is found or the list is exhausted without finding a match.

5 Greedy Algorithms

- It is used to solve problems that have ‘n‘ inputs and require us to obtain a subset that satusfies some constariants.
- Any subset taht satisfied the constraints is called as feasible solution.
- We need to find the optimum feasible solution.

Example: A Greedy Algorithm for Coin Changing

1. Set remval = initial-value
2. Choose largest coin that is less than remval.
3. Addcoin to ser of coins and set remval: = rervamal-coin-value.
4. repeat steps 2 and 3 until remval = 0;

5.1 Knapsack Problem

- An optimization problem is one in which you want to find, not just a solution, but the best solution.
- A “greedy algorithm“ sometimes works well for optimization problems.
- A greedy algorithm works in phases. At each phase:
  - You take the best you can get right now, without regard for future consequences.
  - You hope that by choosing a local optimum at each step, you will end up at a global optimum.