COL351: Analysis and Design of Algorithms

Ragesh Jaiswal, CSE, IITD

Introduction

Introduction

- Algorithm: A step-by-step way of solving a problem.
- Design of Algorithms:
  - *"Algorithm is more of an art than science"*
  - However, we will learn some basic tools and techniques that have evolved over time. These tools and techniques enable you to effectively design and analyse algorithms.
- Analysis of Algorithms:
  - Proof of correctness: An argument that the algorithm works correctly for all inputs.
    - Proof: A valid argument that establishes the truth of a mathematical statement.
  - Analysis of worst-case running time as a function of the input size.

Introduction

- Proof: A valid argument that establishes the truth of a mathematical statement.
    - The statements used in a proof can include axioms, definitions, the premises, if any, of the theorem, and previously proven theorems and uses rules of inference to draw conclusions.
- A proof technique very commonly used when proving correctness of Algorithms is *Mathematical Induction*.

Definition (Strong Induction)

To prove that $P(n)$ is true for all positive integers, where $P(n)$ is a propositional function, we complete two steps:

- Basis step: We show that $P(1)$ is true.
- Inductive step: We show that for all k, if $P(1), P(2), ..., P(k)$ are true, then $P(k+1)$ is true.

Introduction

Definition (Strong Induction)

To prove that $P(n)$ is true for all positive integers, where $P(n)$ is a propositional function, we complete two steps:

- Basis step: We show that $P(1)$ is true.
- Inductive step: We show that for all k, if $P(1), P(2), ..., P(k)$ are true, then $P(k+1)$ is true.

- Question: Show that for all $n > 0$, $1 + 3 + ... + (2n-1) = n^2$.

Introduction

- Question: Show that for all $n > 0$, $1 + 3 + ... + (2n - 1) = n^2$.

Proof

- Let $P(n)$ be the proposition that $1 + 3 + 5 + ... + (2n - 1)$ equals n^2.
- Basis step: $P(1)$ is true since the summation consists of only a single term 1 and $1^2 = 1$.
- Inductive step: Assume that $P(1), P(2), ..., P(k)$ are true for any arbitrary integer k. Then we have:

$$\begin{aligned} 1 + 3 + ... + (2(k+1) - 1) &= 1 + 3 + ... + (2k - 1) + (2k + 1) \\ &= k^2 + 2k + 1 \quad (\text{since } P(k) \text{ is true}) \\ &= (k+1)^2 \end{aligned}$$

  This shows that $P(k + 1)$ is true.
- Using the principle of Induction, we conclude that $P(n)$ is true for all $n > 0$. $\square$

Introduction

- Algorithm: A step-by-step way of solving a problem.
- Design of Algorithms:
  - *"Algorithm is more of an art than science"*
  - However, we will learn some basic tools and techniques that have evolved over time. These tools and techniques enable you to effectively design and analyse algorithms.
- Analysis of Algorithms:
  - Proof of correctness: An argument that the algorithm works correctly for all inputs.
    - Proof: A valid argument that establishes the truth of a mathematical statement.
  - Analysis of worst-case running time as a function of the input size.

Introduction

- Algorithm Design Techniques
  - Divide and Conquer
  - Greedy Algorithms
  - Dynamic Programming
  - Network Flows

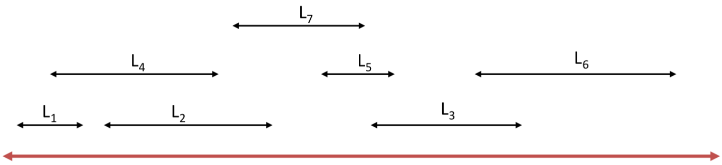
Introduction

- Material that will be covered in the course:
  - Basic graph algorithms
  - Algorithm Design Techniques
    - Divide and Conquer
    - Greedy Algorithms
    - Dynamic Programming
    - Network Flows
  - Computational intractability

- Some examples of Divide and Conquer Algorithms:
    - Binary Search
    - Median finding
    - Multiplying numbers
    - Merge sort, quick sort.

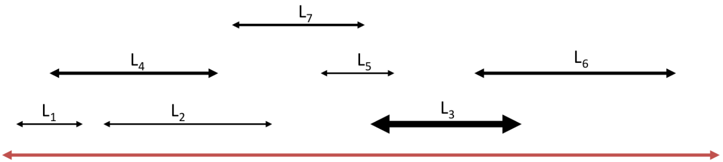
Problem

Interval scheduling: You have a lecture room and you get n requests for scheduling lectures. Each request has a start time and an end time. The goal is to maximise the number of lectures.



Problem

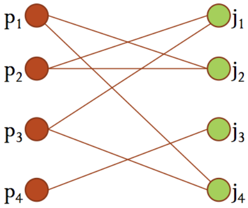
Interval scheduling: You have a lecture room and you get n requests for scheduling lectures. Each request has a start time, an end time, and a price (that you will get in case the lecture is scheduled). The goal is to maximise your earnings.



Problem

Job assignment: There are n people and n jobs. Each person has a list of jobs he/she could possibly do. Find a job assignment so that:

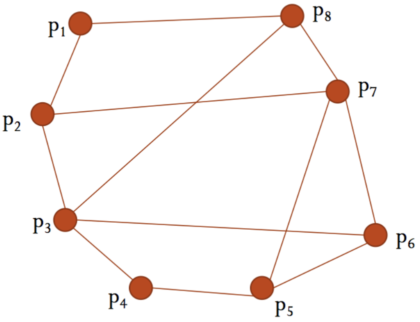
1. each job is assigned to a different person, and
2. each person is assigned a job from his/her list.



- Is it always possible to find a fast algorithm for any problem?

Problem

Given a social network, find the largest subset of people such that no two people in the subset are friends.



- The problem in the previous slide is called the Independent Set problem and no one knows if it can be solved in polynomial time (quickly).
- There is a whole class of problems to which Independent Set belongs.
- If you solve one problem in this class quickly, then you can solve all the problems in this class quickly.
- You can also win a million dollars!!
- We will see techniques of how to show that a new problem belongs to this class:
    - Why: *because then you can say to your boss that the new problem belongs to the difficult class of problems and even the most brilliant people in the world have not been able to solve the problem so do not expect me to do it. Also, if I can solve the problem there is no reason for me to work for you!*

End