CSL 356: Analysis and Design of Algorithms

Instructor: Ragesh Jaiswal

CSE, IIT Delhi

Administrative stuff

Administrative stuff

- Instructor:
  - Ragesh Jaiswal
  - *Office*: 403, SIT Building
  - *Email*: rjaiswal@cse.iitd.ac.in
- Teaching Assistants:
  - Syamantak Das
  - Neha Sengupta
  - Sushant Saxena
  - Vivek Mittal

- Send email to set up a meeting with me or the TAs.

Administrative stuff

- **Lectures:**
  - *Location*: VI LT1
  - *Time*: Tuesday, Wednesday, and Friday, 10:00 - 10:50am.

- Tutorials:
  - *Location*: IIA 101
  - *Time*: TBD.
  - *No Tutorial in the first week.*

Administrative stuff

- **Grading Scheme:**
  - *Homeworks*: 6 homeworks, 5 points each.
  - *Minor 1 and 2*: 20 points each.
  - *Major*: 30 points.

- Policy on cheating:
  - Anyone found using unfair means in the course will receive an **F** grade.
  - You must write the homework solution on your own.

Administrative stuff

- Books:
  - *Algorithm Design* by Jon Kleinberg and Eva Tardos.
    - I will be closely following this book for the course. So, try to get access to a copy of this book.
  - *Introduction to Algorithms* by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Cliff Stein.

- Course webpage:
  - www.cse.iitd.ac.in/~rjaiswal/2014/csl356
    - The page will contain course information, references, homework problems, tutorial problems and announcements. Please check this page regularly.

Recap.

Data Structures

Recap.

- Growth rates:
  - Arrange the following functions in ascending order of growth rate:
    - n
    - $2^{\sqrt{\log(n)}}$
    - $n^{\log(n)}$
    - $2^{\log(n)}$
    - $\frac{n}{\log(n)}$
    - n^n

Recap.

- Problem: Given a positive integer check if the number is prime
- Solution:

```
isPrime(n){
    for i = 2 to (n-1){
        if (n % i== 0)then output(“no”)
    }
    output(“yes”)
}
```

- What is the running time of this algorithm?

Recap.

- Problem: Given a sorted array, check if it contains a given number n.
- Solution: binary search
- Running time?
- Alternate solution: Divide the array into 3 parts and recursively find the element in one of the three parts.
- Running time?

Recap.

- Problem: Given two n bit numbers, multiply them.
- What is the running time of the naïve algorithm?
- Is it possible to get a better algorithm?

Algorithms

Introduction

Introduction

- Algorithm: A step-by-step way of solving a problem.
- *"Algorithm is more of an art than science"*.
- However, we will learn some basic tools and techniques that have evolved over time using which you can effectively design and analyze algorithms.

Introduction: Techniques

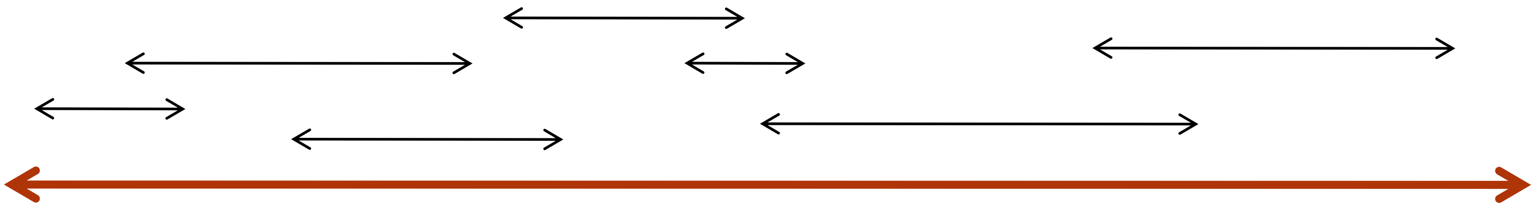
- Divide and Conquer
- Greedy Algorithms
- Dynamic Programming
- Network Flows

Introduction: Divide and Conquer

- Binary search
- Median Finding
- Multiplying numbers
- Merge sort, quick sort.

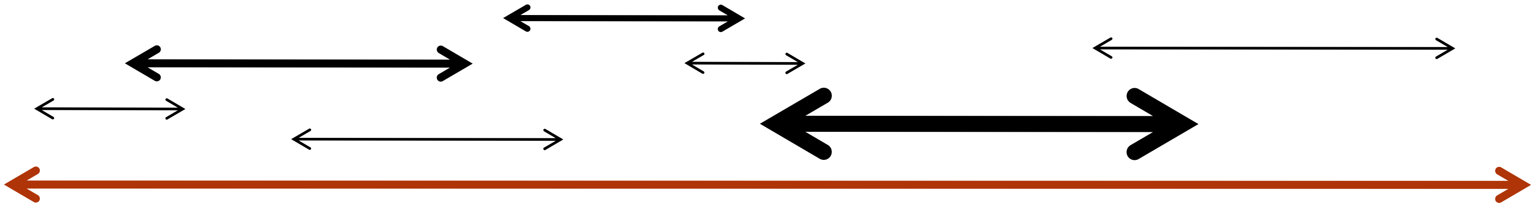
Introduction: Greedy algorithms

- Problem:
    - *Interval scheduling*: You have a lecture room and you get n requests for scheduling lectures. Each request has a start and end time. Goal is to maximize the number of lectures?



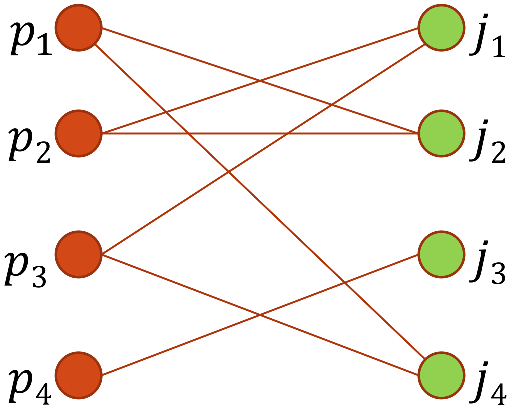
Introduction: Dynamic Programming

- Problem:
  - *Interval scheduling*: You have a lecture room and you get n requests for scheduling lectures. Each request has a start time, end time, and a price (in case it is scheduled). Goal is to maximize your earnings?



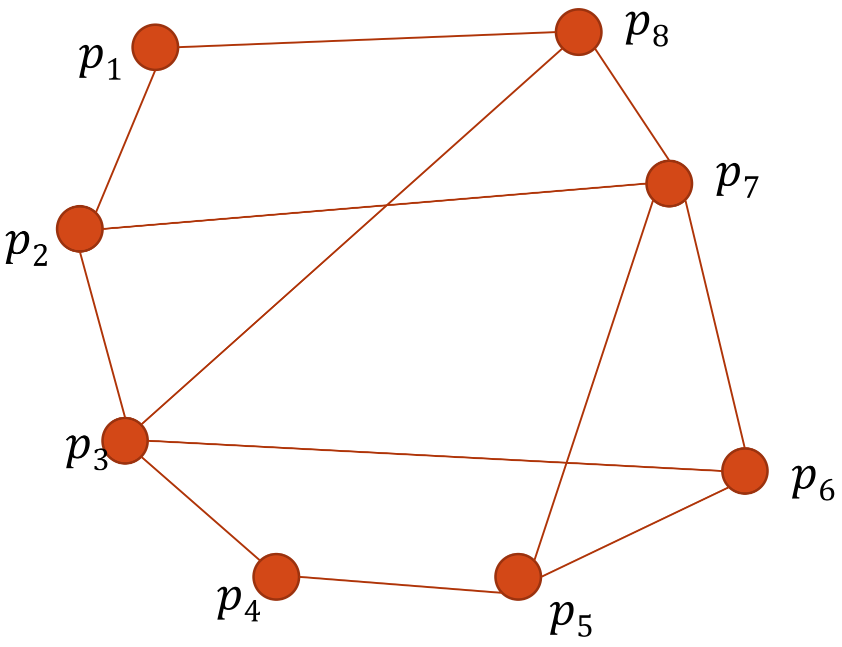
Introduction: Network flows

- Problem:
  - *Assigning jobs*: There are n people and n jobs. Each person has list of jobs he could possibly do. Find a job assignment so that:
    - each job assigned assigned to a different person, and
    - each person is assigned a job from his list.



Introduction: Is it always possible to find an algorithm for a problem

- Computational Intractability:
  - Problem: Given a friend network find the largest subset of people such that no two people in the subset are friends.



Introduction: Computational Intractability

- The problem is called the independent set problem 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

Problems to think about:

1. In the interval scheduling with weights problem (slide 17) suppose you pick the request with the largest value, then pick the request with the next highest value and that does not conflict the request already chosen and so on. Does this strategy always maximizes the scheduler's profit?