Discrete Mathematics: Introduction

Slides by: Christopher M. Bourke
Instructor: Berthe Y. Choueiry

Spring 2006

Computer Science & Engineering 235
Introduction to Discrete Mathematics
cse235@cse.unl.edu

Notes

Computer Science & Engineering 235

Discrete Mathematics

- Roll
- Syllabus
- Lectures: M/W/F 12:30 – 1:20 (Avery 109)
- Recitations: Mondays 4:30 – 5:20 (Avery 118)
- Office hours:
  - Instructor: M/W 3:30 – 4:30 (Avery 123B)
  - TA: xxx M/W 3:30 – 4:30 (Avery 123C)
- Must have cse account
- Must use webhandin
- Bonus points: report bugs

Notes

Why Discrete Mathematics? I

You have to.

Computer Science is *not* programming.

Its not even Software Engineering.

> *"Computer Science is no more about computers than astronomy is about telescopes." –Edsger Dijkstra*

Computer Science is *problem solving*.

Notes

Why Discrete Mathematics? II

Mathematics is at the heart of problem solving.

Often, even *defining* a problem requires a level of mathematical rigor.

Competent use and analysis of models/data structures/algorithms requires a solid foundation in mathematics.

Justification for why a particular way of solving a problem is *correct* or *efficient* (i.e. better than another way) requires analysis within a well defined mathematical model.

Notes

Why Discrete Mathematics? III

Abstract thinking is necessary to applying knowledge.

Rarely will you encounter a problem in an abstract setting (your boss is not going to ask you to solve MST). Rather, it is up to you to determine the proper model of such a problem.

Notes

Scenario I

A limo company has hired you (or your company) to write a computer program to automate the following tasks for a large event.

Task 1 – In the first scenario, businesses are request limos and drivers for a fixed period of time (specifying a start-date/time and end-date/time) and charged a flat rate. The program should be able to generate a schedule so that the maximum number of customers can be accommodated.

Notes

Scenario II

Task 2 – In the second scenario, the limo service is considering allowing customers to *bid* on a driver (so that the highest bidder gets a limo/driver when there aren't enough available). The program should thus make a schedule a feasible (i.e. no limo can handle two customers at the same time) while at the same time, maximizing the profit by selecting the highest *overall* bids.

Notes

Scenario III

Task 3 – In a third scenario, a customer is allowed to specify a *set* of various times and bid an amount for the entire event. A driver must choose to accept the entire set of times or reject it all. The scheduler must still maximize the profit.

Notes

Scenario

What's your solution?

How can you *model* such scenarios?

How can you develop algorithms for these scenarios?

How can you justify that that they work? That they actually guarantee an optimal (i.e. maximized profit) solution?

Notes

Scenario

The fundamentals that this course will teach you are the foundations that you will use to eventually solve these problems.

The first scenario is easily (i.e. efficiently) solved by a *greedy algorithm*.

The second scenario is also efficiently solvable, but by a more involved technique, *dynamic programming*.

The last scenario is not efficiently solvable (it is NP-complete) by any known technique. It is believed that to guarantee an optimal solution, one needs to look at all (exponentially many) possibilities.

Notes

Fundamentals
Notation

A *set* is a collection of similar objects. We denote a set using brackets. For example,

$$S = \{s_1, s_2, s_3, \ldots, s_n\}$$

is a finite set and

$$S = \{s_1, s_2, s_3, \ldots\}$$

is an infinite set.

We denote that an object is an *element* of a set by the notation,

$$s_1 \in S$$

read "s_1 (is) in S" (or we can write $s_1 \notin S$ for "s_1 (is) *not* in S")

Notes

Fundamentals
Notation

You should at least be familiar with the sets of integers, rationals and reals.

- We denote the set of *natural numbers* as

$$\mathbb{N} = \{0, 1, 2, 3, \ldots\}$$

- We denote the set of *integers* as

$$\mathbb{Z} = \{0, 1, -1, 2, -2, 3, -3, \ldots\}$$

- We denote the set of *rational* numbers as

$$\mathbb{Q} = \left\{\frac{a}{b} \mid a, b \in \mathbb{Z}\right\}$$

- We denote the set of *reals* as

$$\mathbb{R} = \{x \mid x \text{ is a decimal number}\}$$

Notes

Algebra I

Definition

Let $a, b \in \mathbb{Z}$ with $b \neq 0$. we say that b *divides* a if and only if

$$a = qb$$

for some integer q. We will use the notation

$$b \mid a$$

Notes

Algebra II

Example

2 divides 64 since

$$64 = (32)2$$

3 divides 27 since

$$27 = (9)3$$

However, 2 does not divide 27 since there is no integer q such that

$$27 = 2q$$

In this case, we write $2 \nmid 27$

Notes

Topics

Topic	Sections
Propositional Logic	1.1 - 1.2
Predicate Logic	1.3 - 1.4
Proofs	1.5
Sets	1.6 - 1.7
Functions	1.8
Relations	7.1, 7.3 - 7.6
Algorithms	2.1 - 2.5
Induction	3.1 - 3.3
Counting	4.1 - 4.2
Combinatorics	4.3 - 4.5
Recursion	6.1 - 6.2
PIE	6.5
Graphs	8.1 - 8.5
Trees	9.1 - 9.3

Notes