GREATER JAKARTA • BANDUNG • MALANG

Introduction to Scientific Computing

Week 1

Chapter 0

S. Pradono Suryodiningrat

People
Innovation
Excellence

Course Introduction

No.	Components	Percentage		Learning Outcomes
1.	Class Participation	10	%	LO 1, LO 2, LO 3, LO 4, LO 5
2.	Assignment	30	%	LO 1, LO 2, LO 3, LO 4, LO 5
3.	Mid Examination	30	%	LO 1, LO 2, LO 3, LO 4
4.	Final Examination	30	%	LO 1, LO 2, LO 3, LO 4, LO 5
	Total	100	%	

Book

- Sipser, M, (2013). Introduction to the Theory of Computation (3^{rd} Ed). Cengage Learning. ISBN: 978-1-133-18779-0

0.1 – Automata, Computability, and Complexity

- What are the fundamental capabilities and limitations of computers?
    - Complexity
    - Computability
    - Automata

Complexity Theory

What makes some problems computationally hard and others easy?

1. By understanding which aspect of the problem is at the root of the difficulty, you may be able to alter it so that the problem is more easily solvable
2. You may be able to settle for less than a perfect solution to the problem
3. Some problems are hard only in the worst-case situation, but easy most of the time. Depending on the application, you may be satisfied with a procedure that occasionally is slow but usually runs quickly
4. You may consider alternative types of computation, such as randomized computation, that can speed up certain tasks

Computability Theory

- First half of the twentieth century
- Kurt Godel, Alan Turing, and Alonzo Church
- Basic problems cannot be solved by computers
  - True or false
  - Bread and butter of mathematicians
- In complexity theory
  - the objective is to classify problems as easy ones and hard ones; whereas in computability theory, the classification of problems is by those that are solvable and those that are not.
- Computability theory introduces several of the concepts used in complexity theory.

Automata Theory

- Deals with the definitions and properties of mathematical models of computation
- Finite automata
  - used in text processing, compilers, and hardware design
- context-free grammar
  - used in programming languages and artificial intelligence

0.2 – Mathematical Notion and Terminology

- Sets
  - A group of objects represented as a unit
  - May contain any type of object, including numbers, symbols, and even other sets
  - The objects in a set are called its elements or members

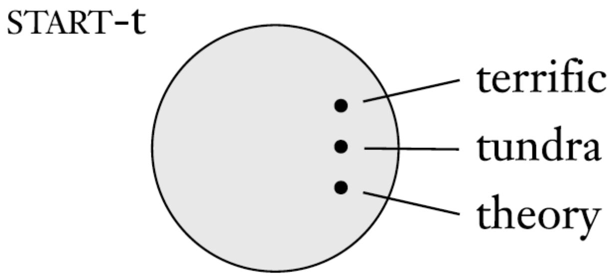
$$S = \{7, 21, 57\}$$

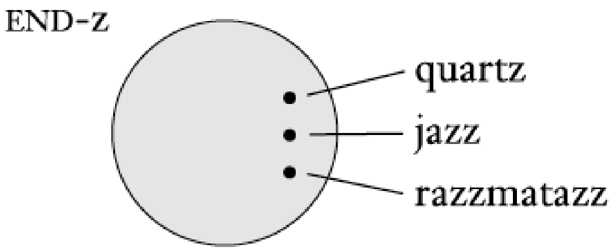
$$S = \{7, 21, 57\}$$

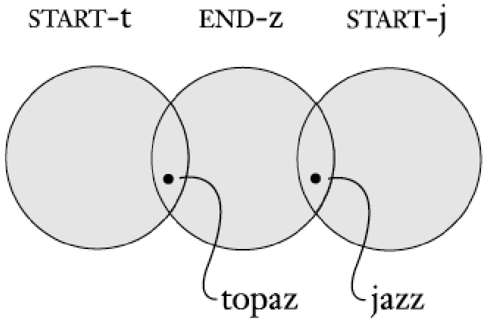
- The sets contains the elements 7, 21, 57
- Symbols ∈ and ∉ denote set membership and non-membership
- We write $7 \in \{7, 21, 57\}$ and $8 \notin \{7, 21, 57\}$
- A is a subset of B, written $A \subseteq B$, if every member of A also is a member of B
- A is a proper subset of B, written A!⊆B, if A is a subset of B and not equal to B

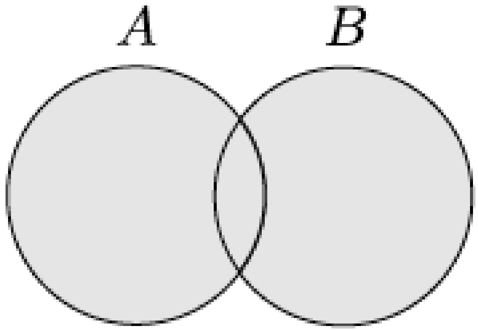
- The set with zero members is called the empty set and is written ∅.
- A set with one member is sometimes called a singleton set
- a set with two members is called an unordered pair
- two sets A and B, the **union** of A and B, written AUB, is the set we get by combining all the elements in A and B into a single set.
- The **intersection** of A and B, written A ∩ B, is the set of elements that are in both A and B.
- The complement of A, written $\overline{A}$, is the set of all elements under consideration that are not in A.

- For sets, we use a type of picture called a Venn diagram

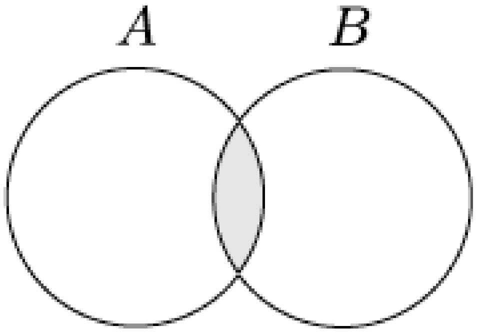








(a)



(b)

- Sequences and Tuples
  - objects is a list of these objects in some order
  - (7, 21, 57)
  - The order doesn’t matter in a set, but in a sequence it does
- Tuples
  - Finite sequences often are
  - A sequence with k elements is a k-tuple
  - Thus (7, 21, 57) is a 3-tuple
  - A 2-tuple is also called an ordered pair

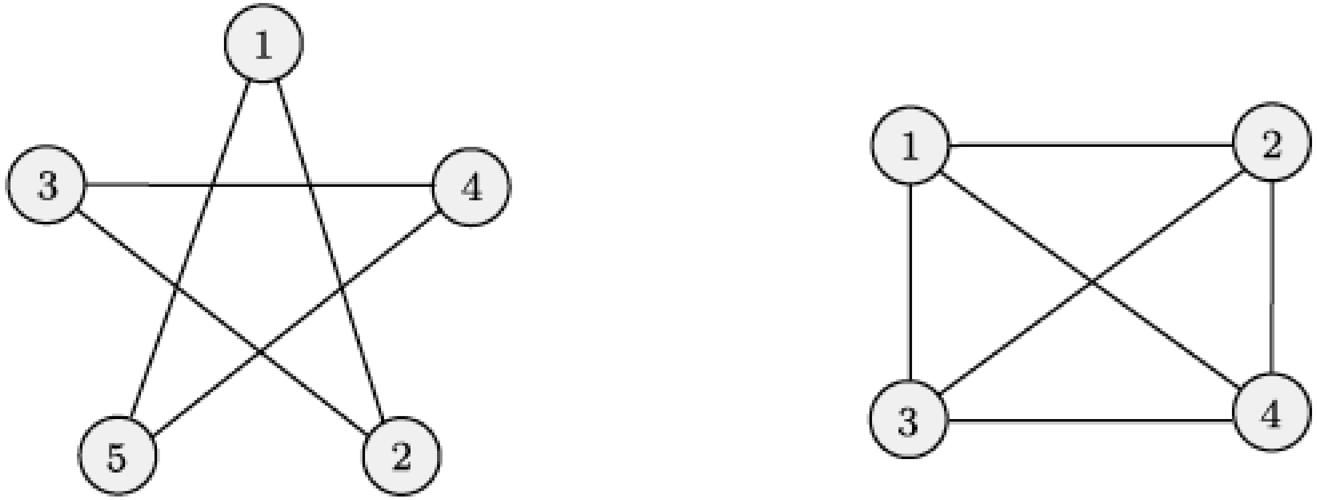
- Tuples
  - Cartesian product or cross product
  - If A = {1, 2} and B = {x, y, z},
  - A x B = { (1, x), (1, y), (1, z), (2, x), (2, y), (2, z) }.
  - A x B x A = { (1, x, 1), (1, x, 2), (1, y, 1), (1, y, 2), (1, z, 1), (1, z, 2), (2, x, 1), (2, x, 2), (2, y, 1), (2, y, 2), (2, z, 1), (2, z, 2) }

$$\overbrace{A \times A \times \cdots \times A}^{k} = A^k.$$

- Functions
  - an object that sets up an input–output relationship
  - A function takes an input and produces an output
  - If f is a function whose output value is b when the input value is a, we write $f(a) = b$
  - Mapping
  - The set of possible inputs to the function is called its domain.
  - The outputs of a function come from a set called its range.
  - The notation for saying that f is a function with domain D and range R is

$$f : D \rightarrow R$$

- Graphs
  - a set of points with lines connecting some of the points



(a) (b)

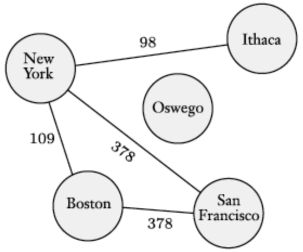


FIGURE 0.13
Cheapest nonstop airfares between various cities

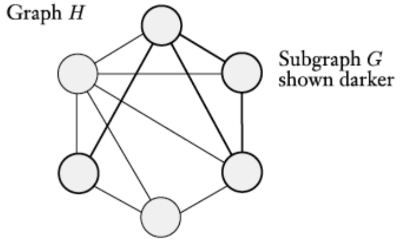


FIGURE 0.14
Graph G (shown darker) is a subgraph of H

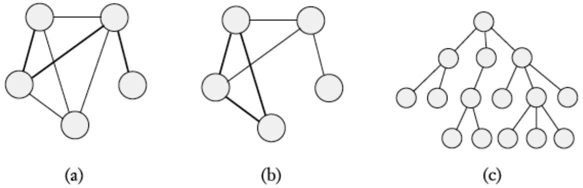


FIGURE 0.15
(a) A path in a graph, (b) a cycle in a graph, and (c) a tree

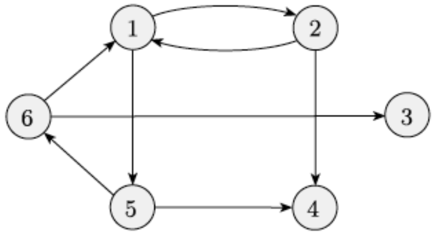


FIGURE 0.16
A directed graph

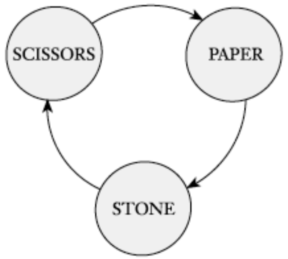


FIGURE 0.18
The graph of the relation *beats*

- Strings
    - are fundamental building blocks in computer science
    - Alphabet
    - Symbos

- Boolean Logic
  - a mathematical system built around the two values TRUE and FALSE

AND	OR	NOT
$0 \wedge 0 = 0$	$0 \vee 0 = 0$	$\neg 0 = 1$
$0 \wedge 1 = 0$	$0 \vee 1 = 1$	$\neg 1 = 0$
$1 \wedge 0 = 0$	$1 \vee 0 = 1$	
$1 \wedge 1 = 1$	$1 \vee 1 = 1$	

- Boolean Logic

Exclusive OR (XOR)	Equality	Implication
$0 \oplus 0 = 0$	$0 \leftrightarrow 0 = 1$	$0 \rightarrow 0 = 1$
$0 \oplus 1 = 1$	$0 \leftrightarrow 1 = 0$	$0 \rightarrow 1 = 1$
$1 \oplus 0 = 1$	$1 \leftrightarrow 0 = 0$	$1 \rightarrow 0 = 0$
$1 \oplus 1 = 0$	$1 \leftrightarrow 1 = 1$	$1 \rightarrow 1 = 1$

0.3 Definition, Theorems, And Proofs

- Definitions
  - describe the objects and notions that we use.
  - may be simple, as in the definition of set given earlier in this chapter, or complex as in the definition of security in a cryptographic system.
  - Precision is essential to any mathematical definition. When defining some object, we must make clear what constitutes that object and what does not.

- Theorem
  - a mathematical statement proved true.
  - Occasionally we prove statements that are interesting only because they assist in the proof of another, more significant statement called lemmas.
  - Occasionally a theorem or its proof may allow us to conclude easily that other, related statements are truecalled corollaries of the theorem.
- Proof
  - is a convincing logical argument that a statement is true

- Proof by construction
  - Prove such a theorem is by demonstrating how to construct the object
- Proof by contradiction
  - we assume that the theorem is false and then show that this assumption leads to an obviously false consequence
- Proof by Induction
  - is an advanced method used to show that all elements of an infinite set have a specified property

Cheers!