Week 1 (4) – Fundamental Data Structures

CST370 – Design & Analysis of Algorithms

Dr. Byun

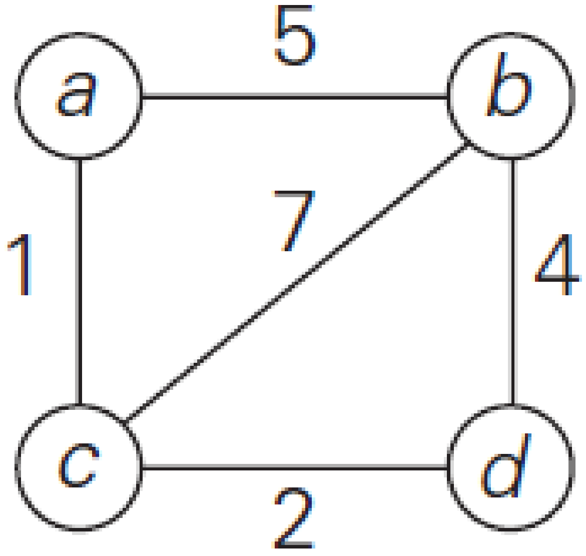
Computer Science

Lecture Objectives

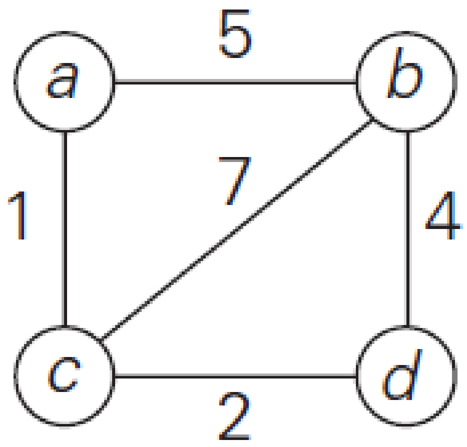
- After completion of this lecture, you will be able to
  - understand the basic terminologies of weighted graphs and trees in computer science.

Weighted Graphs

- Watch this video first
  - https://youtu.be/9r2yyJgFQxU
- A weighted undirected graph and a weighted directed graph are graphs with numbers (or cost) assigned to their edges.
- Example



Weighted Graphs Representation – Adjacency Matrix



→

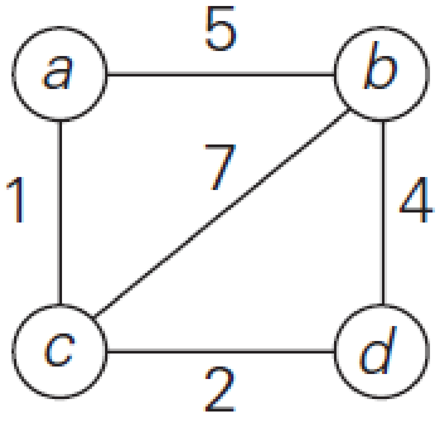
$$\begin{array}{c|cccc} & a & b & c & d \\ a & \infty & 5 & 1 & \infty \\ b & 5 & \infty & 7 & 4 \\ c & 1 & 7 & \infty & 2 \\ d & \infty & 4 & 2 & \infty \end{array}$$

→

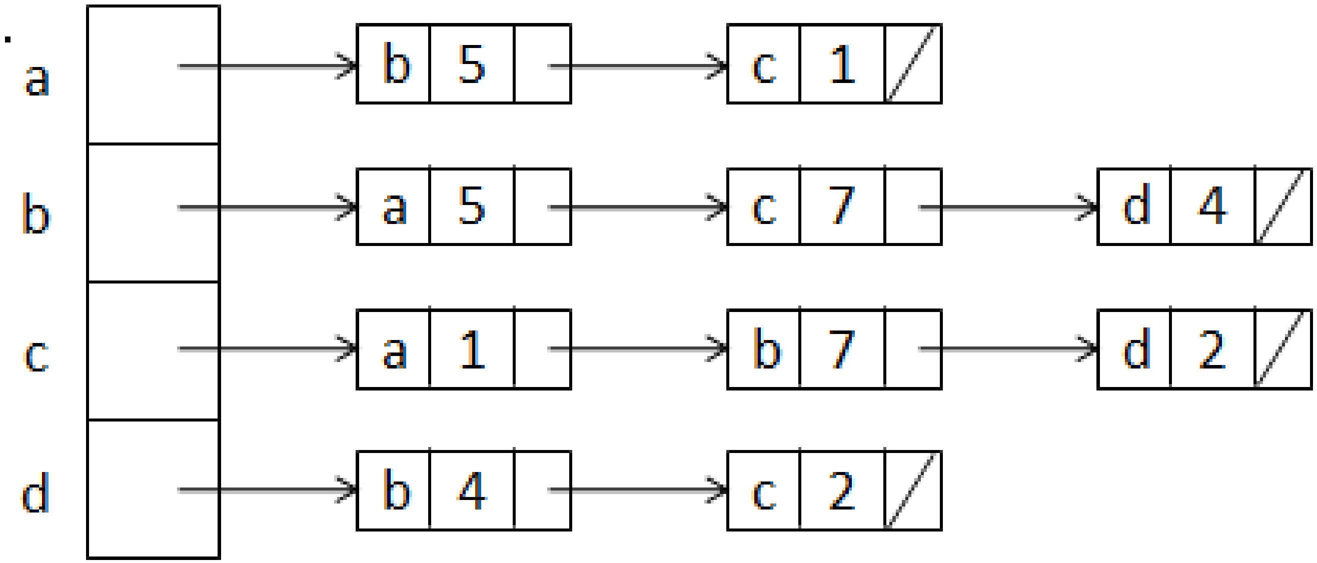
$$\begin{array}{c|cccc} & a & b & c & d \\ a & 0 & 5 & 1 & \infty \\ b & 5 & 0 & 7 & 4 \\ c & 1 & 7 & 0 & 2 \\ d & \infty & 4 & 2 & 0 \end{array}$$

// You can put 0's on the main diagonal.

Weighted Graphs Representation – Adjacency List



➔



a → b 5 → c 1

b → a 5 → c 7 → d 4

c → a 1 → b 7 → d 2

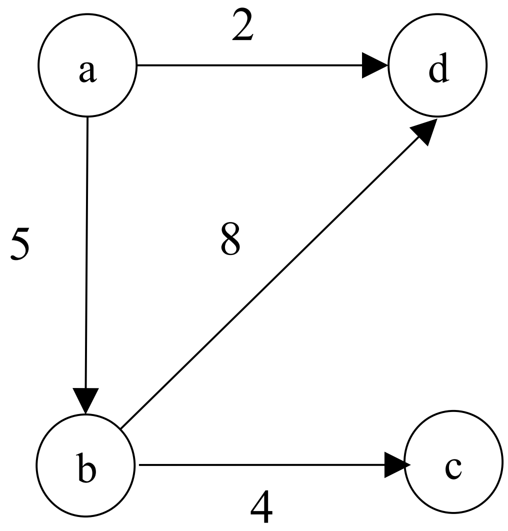
d → b 4 → c 2

<<< Course Instruction >>>

- **Read pages 30 – 31** (before “Trees”) in the textbook before moving on to the next slide.
  - There are many **terminologies** of graphs in the section.
  - Read the section carefully to get the accurate meaning of the terminologies.

Exercise

- Represent the graph in adjacency matrix and adjacency list.



Important Note

- **Do not see the solution immediately** on the next page.

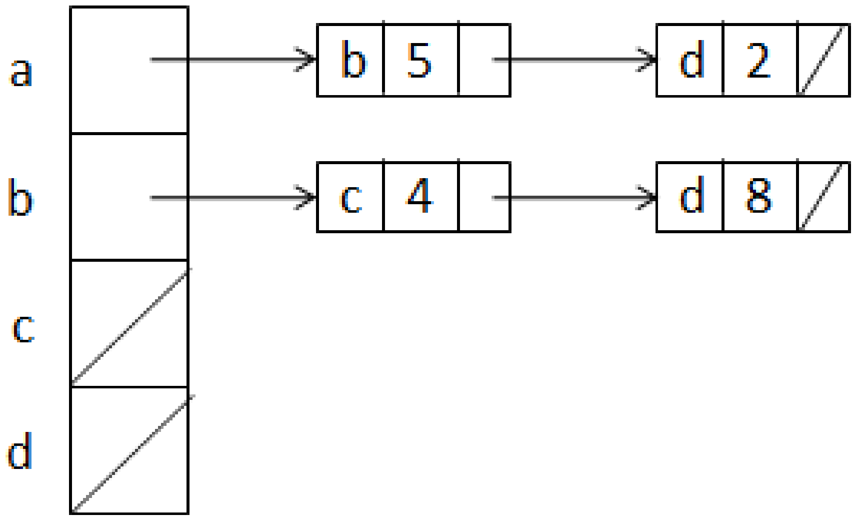
Solution (1 of 2)

- Adjacency matrix

$$\begin{array}{c|cccc|}
 & a & b & c & d \\ \hline
a & \infty & 5 & \infty & 2 \\
b & \infty & \infty & 4 & 8 \\
c & \infty & \infty & \infty & \infty \\
d & \infty & \infty & \infty & \infty \\ \hline
\end{array}
\qquad \text{OR} \qquad
\begin{array}{c|cccc|}
 & a & b & c & d \\ \hline
a & 0 & 5 & \infty & 2 \\
b & \infty & 0 & 4 & 8 \\
c & \infty & \infty & 0 & \infty \\
d & \infty & \infty & \infty & 0 \\ \hline
\end{array}$$

Solution (2 of 2)

- Adjacency list



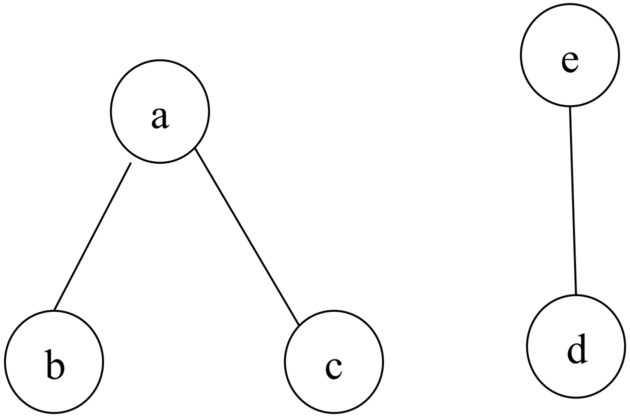
- Note our convention of alphabetical order.
  - For example, the vertex *a* has two edges to the vertexes *b* and *d*. When we represent it, we put the vertex *b* first and then put the vertex *d*.

Paths and Cycles

- Recall the terminologies mentioned in the textbook such as
  - Path
  - Length
  - Simple path
  - Connected
  - Connected component
  - Cycle
  - Acyclic graph

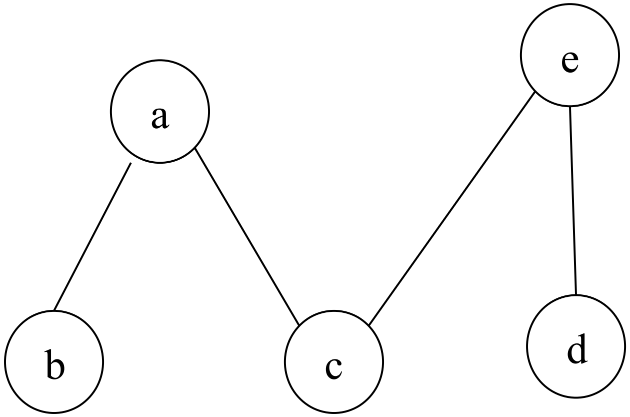
Question

- Is the graph **connected**? **No.**



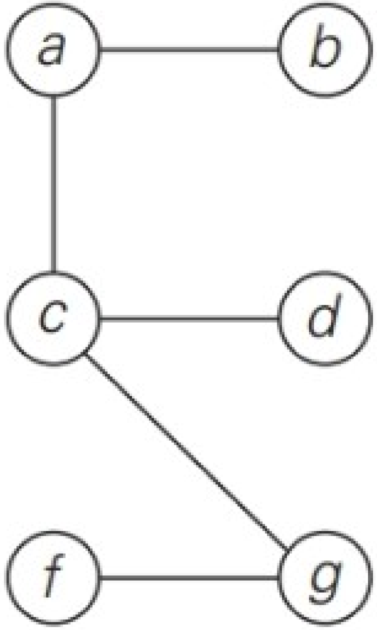
Question

- Is the graph **connected**? **Yes.**



Trees

- Watch this video first
  - https://youtu.be/3n09brVyINI
- Note that a tree is actually a graph (= **connected acyclic graph**).
- Example



Free Tree, Rooted Tree, and Ordered Tree

- There are several different trees in computer science theory such as (free) trees, rooted trees, and ordered trees.
  - Our main interest in the class is the **binary tree** and **binary search tree**.
  - However, it would be great for you to know the general terminologies of trees mentioned in the textbook as a computer scientist.

<<< Course Instruction >>>

- **Read pages 31 – 35** (before “Sets and Dictionaries”) in the textbook before moving on to the next slide.
  - You don’t need to understand the **“first child-next sibling representation”** in the textbook page 35. If you want, you can **skip** that part.

Puzzle: Palindrome Checking

- Let’s assume that you have **a very long character array**.
- How can you determine if the character array is a palindrome or not?
  - Of course, your algorithm should be **efficient**.
- Example
  - racecar // Yes, it’s a palindrome.
  - abcdefghijihgfedcba // Yes, it’s a palindrome.
  - CSUMB //No, it’s not a palindrome.

Important Note

- **Do not see the answer immediately** on the next page.

Solution

- You can use two indexes i and j.
  - i starts from the index 0 (= first character).
  - j starts from the index n-1 (= last character).
  - Check the characters in the index i and j.
  - If they are not the same, return false.
  - Otherwise, i increases by 1 and j decreases by 1 and check again.
  - This way, you can keep checking the characters of i and j until they meet.
  - If they meet, the array is a palindrome.

<<< Course Instruction >>>

- This lesson is over.
  - If you have any questions, please contact your instructor.
- When you are done, study the next lecture (week_1_5.ppt) on the Canvas.