BIM213 - Data Structures and Algorithms

Introduction

About the course

Instructor:	Assist. Prof. Dr. Selcan KAPLAN BERKAYA
Teaching Assistant:	Res. Assist. Emrah DEMIR Res. Assist. Mehmet OZCAN
Class Hours and Location:	09:00-12:00, Friday – B7
Prerequisites:	- BIM101 – Computer Programming I - BIM102 – Computer Programming II
Textbooks:	**Data Structures & Problem Solving Using Java**, Mark Allen Weiss, 4th Edition, Pearson, 2010. **Introduction to Algorithms**, Cormen, Leiserson, Rivest, Stein.

Course Outline

- Iterative algorithms and their analysis
  - Case Study: Iterative Sorting Algorithms
- Recursive algorithm design & analysis
- Lists – ArrayList & LinkedList
- Stacks, Queues
- Trees
- Search Trees
  - Binary Search Trees
  - AVL Trees
  - Splay Trees
- Hash Tables

Attendances

- Recent experiences show that the students who attend the classes are more successful
- All students are responsible for visiting the Mergen page of the course at least two times in each week
- Announcements, assignments, grades, and project subjects will be published on the Mergen system.

Homeworks & Projects

- Grading
  - 1 Midterm – 30%
  - 4 Homeworks – 20%
  - 1 Final – 50%

What's this course about?

- An algorithm (program) is a well-defined computational procedure that
  - takes some values (data) as "input"
  - produces some result as "output"

- Programs receive, manipulate, and output data
  - Need to organize data according to problem being solved
  - Data structures are methods for organizing data

Data Structures (DS): What, How, and Why?

- Data structures are methods for organizing data
- Formal definition of DS: Abstract Data Type (ADT)
  - A "toolkit" of operations for manipulating data
  - E.g. A list with operations insert and delete
  - E.g. A stack with operations push and pop
  - E.g. A queue with operations enqueue and dequeue

Data Structures (DS): What, How, and Why?

- Program design depends crucially on data organization, i.e., how data is structured for use by the program
  - Implementation of some operations becomes easier or harder
  - Speed of program may dramatically decrease or increase
  - Memory used may increase or decrease

- We will see examples of these throughout the course

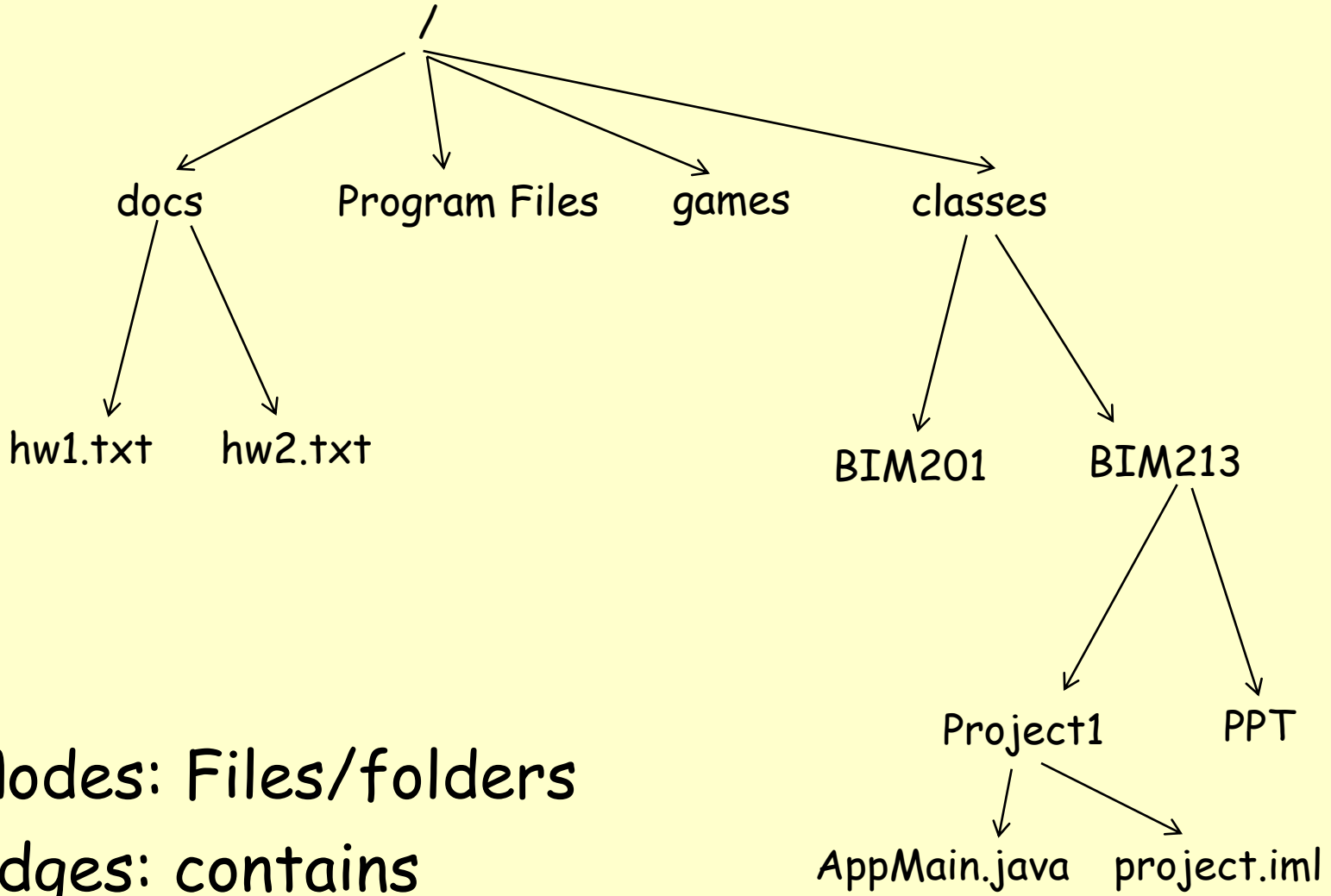
Course Goals for Data Structures

- Study different implementation techniques for some fundamental ADTs
- Learn how to choose the "best" one
- Learn how to modify standard ADTs for specific problems, and create new ADTs

Data Structures are used…

- Everywhere
  - Systems (Operating Systems, Computer Networks)
  - Graphics
  - Databases
  - Theory
  - Artificial Intelligence
  - Information Retrieval
  - …
- Maybe the most important class in your curriculum ☺
  - Guaranteed good and important stuff

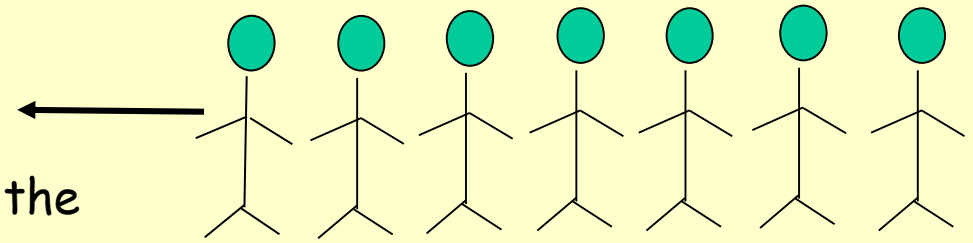
E.g. 1: Tree of Files and Folders



- Nodes: Files/folders
- Edges: contains

E.g. 2: Queue of People

Queue of people waiting to pay bills

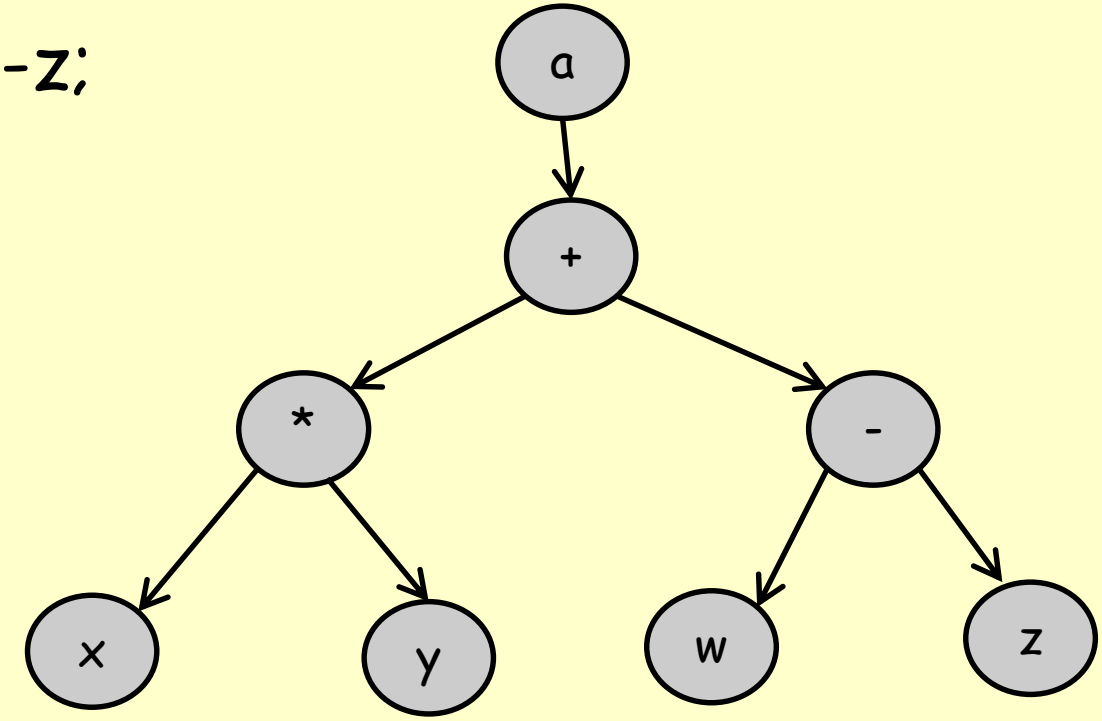


Front of the queue:
Next person to be served

Rear of the queue
Next person will join the queue
from the rear

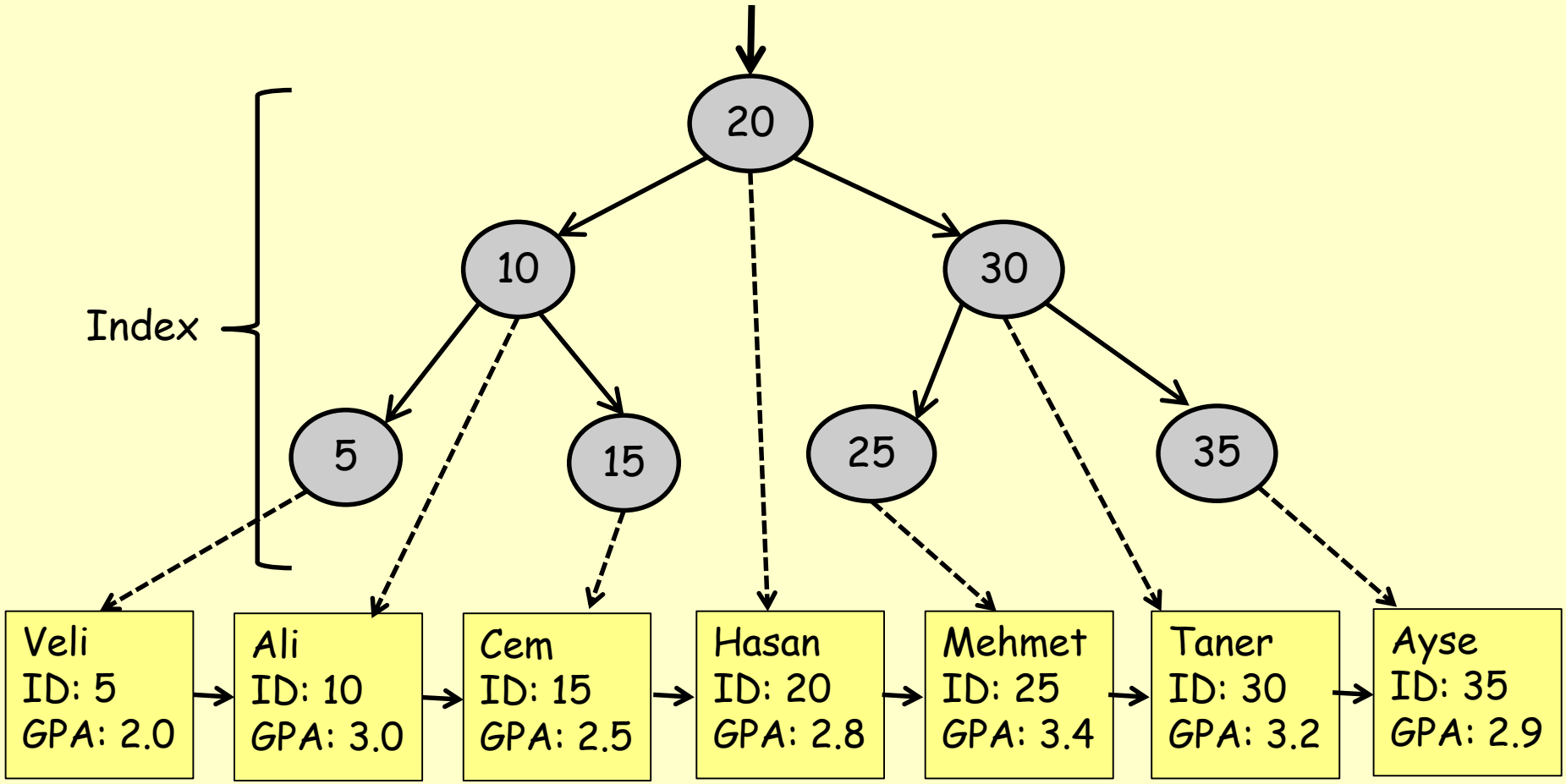
E.g. 3: Representing Expressions

- a = x*y + w-z;



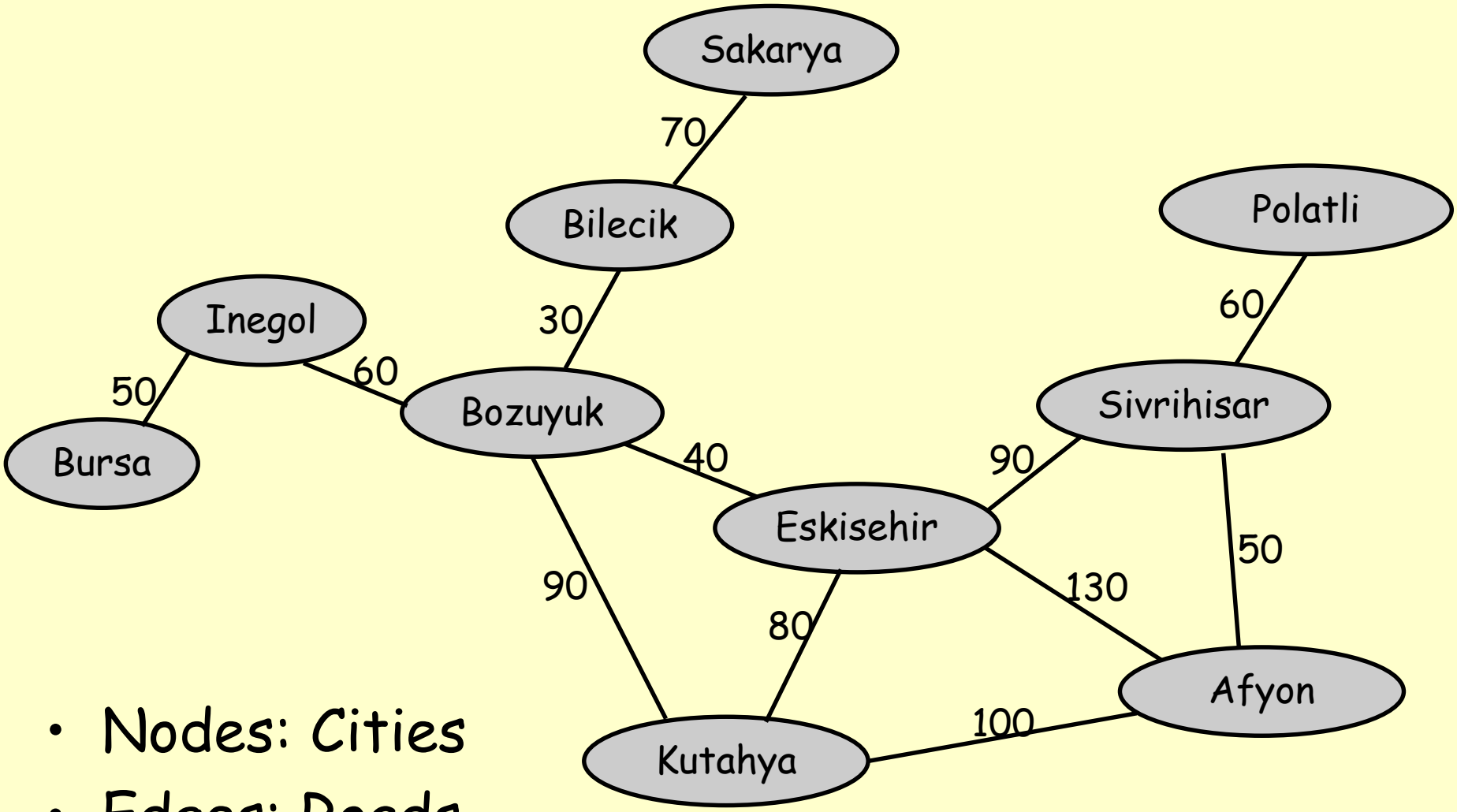
- Nodes: Operands/Operators
- Edges: Relationships

E.g. 4: Balanced Search Trees



- Nodes: (Key/Value) pairs, Edges: Relationships

E.g. 5: Transportation Networks



- Nodes: Cities
- Edges: Roads

Algorithms and Their Analysis

- What is an algorithm?
  - A sequence of steps (a "program") that accomplishes a task
  - Independent of Programming Language

- Many different algorithms may correctly solve a given task
  - But choice of a particular algorithm may have enormous impact on time and memory used
  - Time versus space tradeoffs are very common

Types of Algorithms

- Iterative Algorithms
- Recursive (Divide & Conquer) Algorithms
- Randomized Algorithms
- Dynamic Programming
- Greedy Algorithms
- Approximation Algorithms
- Genetic Algorithms

Course Goals for Algorithms

- Understand the mathematical fundamentals needed to analyze algorithms
- Learn how to compare the efficiency of different algorithms in terms of running time and memory usage
- Study a number of standard algorithms for data manipulation and learn to use them for solving new problems