CSE2010

Introduction to Data Structures

Jungsun Kim

College of Computing
Hanyang University
Spring 2024

Who am I?

김정선 (金 正 善, Jungsun Kim)
한양대학교 컴퓨터학부
소프트웨어기술 및 아키텍처 연구실
(Software Technology &
Architecture Research Lab.)

Office: 4공학관 316호
Email: kimjs@hanyang.ac.kr

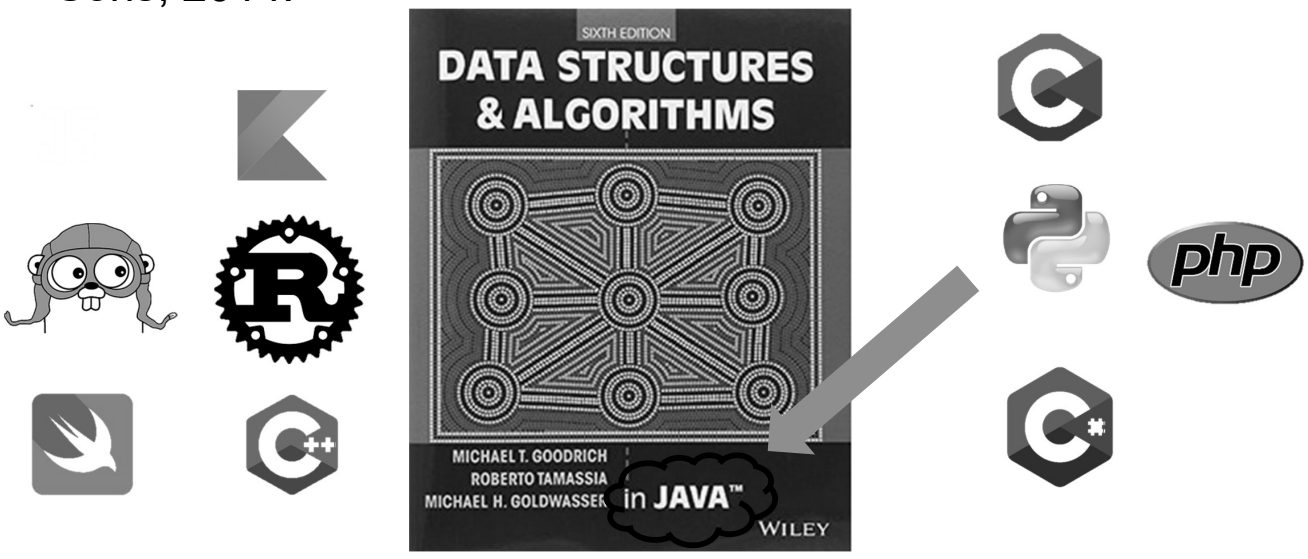
Who is your Grading Assistant?

김정인
한양대학교 인공지능융합학과
모바일통신 및 컴퓨팅 연구실

Office: 제 3공학관 422-2호
Email: rkrkrk24@hanyang.ac.kr
Phone: (031) 400-4757
Mobile: 010-3911-2368

Textbook

Data Structures and Algorithms in Java, 6th ed., Michael Goodrich & Roberto Tamassia, John Wiley & Sons, 2014.



Class Website

http://veenker.hanyang.ac.kr/~jskim/cse2010.html

CSE2010 Introduction to Data Structures

Spring 2024

Syllabus | Announcements | Handouts | Slides | Homeworks | Links

Department of Computer Science and Engineering
Instructor: Jungsun Kim
kimjs@hanyang.ac.kr
(031) 400-5669

	Time	Location
Section A	(Thu) 09:00-10:15	제1공학관 304호
	(Fri) 09:00-10:15	제1공학관 305호
Section B	(Thu) 10:30-11:45	제1공학관 304호
	(Fri) 10:30-11:45	제1공학관 305호

Object

The course is aimed to improve student`s programming techniques by introducing data structures, abstract data types, and algorithms for sorting and searching. Course topics include the design and implementation of data structures such as arrays, lists, stacks, queues, trees, heaps, and graphs. Other topics include the algorithm analysis method and the study of algorithms for manipulating those data structures. On completion of the course, the student will be able to develop efficient applications.

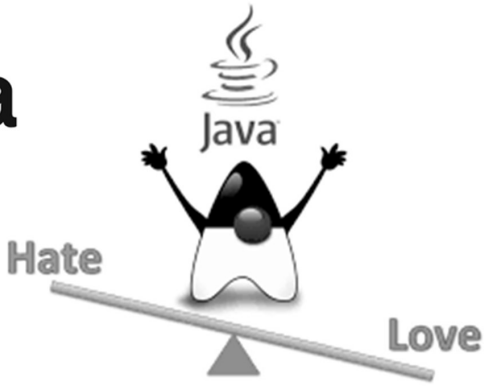
Assumptions and Prerequisites

Basic knowledge of programming

Assumptions and Prerequisites

Strongly recommended

- Experience with the basics of Java language



Assumptions and Prerequisites

Some mathematical backgrounds

Basic Differentiation Formulas

$\frac{dk}{dx} = 0$ *where k = constant*

$\frac{d(x)}{dx} = 1$

$\frac{d(kx)}{dx} = k$ *where k = constant*

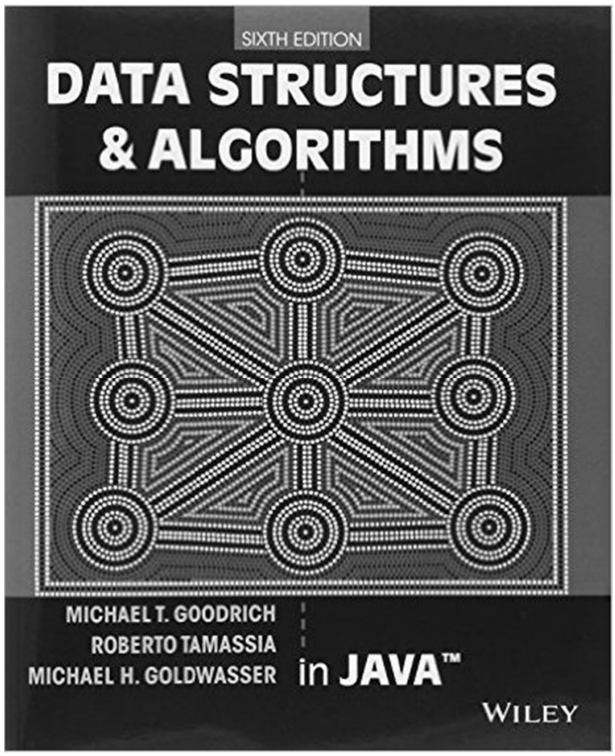
$\frac{d(x^n)}{dx} = nx^{n-1}$

Derivatives of Logarithmic and Exponential functions

$\frac{d}{dx}(\log_a x) = \frac{1}{x \ln a}$ $\quad \frac{d}{dx}(\ln x) = \frac{1}{x}$

$\frac{d}{dx}(a^x) = a^x \ln a$ $\quad \frac{d}{dx}(e^x) = e^x$

Examples on the board.



Grading (Tentative)

Midterm Exam: 35%

Final Exam 45%

Homeworks: 15%

Cheating?

Course Topics

Nicklaus Wirth:

- (Creator of PASCAL Language)

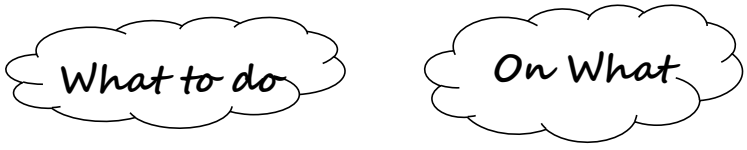
```
Program TowersOfHanoi(input,output);

Var
   disks:   integer;

Procedure Hanoi(source, temp, destination: char;  n: integer);

   begin
   if n > 0 then
      begin
      Hanoi(source, destination, temp, n - 1);
      writeln('Move disk ',n:1,' from peg ',source,' to peg ',destination);
      Hanoi(temp, source, destination, n - 1);
      end;
   end;

begin
write('Enter the number of disks: ');
readln(disks);
writeln('Solution:');
Hanoi('A','B','C',disks);
end.
```


Course Topics

Nicklaus Wirth:

- (Creator of PASCAL Language)

Programs = Algorithms + Data Structures



Course Topics

Algorithms

- Finite sequence of steps to accomplish a particular task in a finite amount of time

  searching, sorting, data structure-specific algorithms, etc.

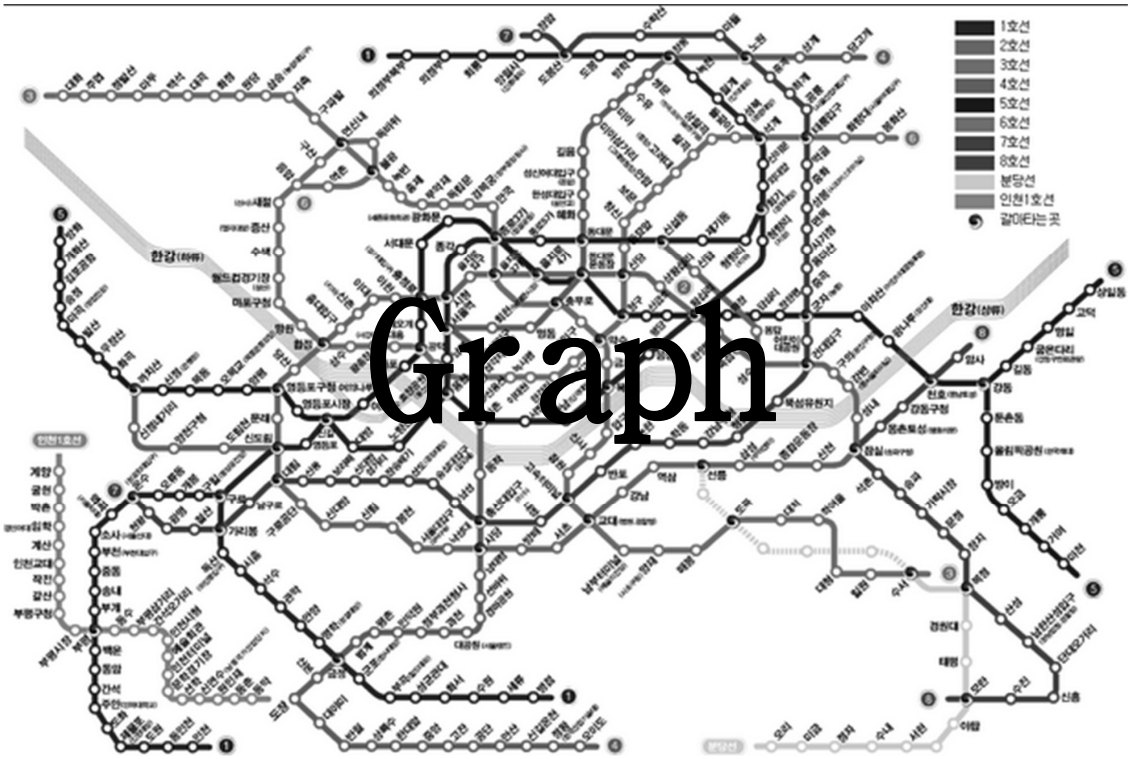
Data structures

- Concrete representation of how data is organized in memory ← *informal definition for now*
- Related to the collections of data

  메모리상에서 체계적으로 정의된 관계를 갖도록
  구성된 데이터 Collection의 표현(구조)

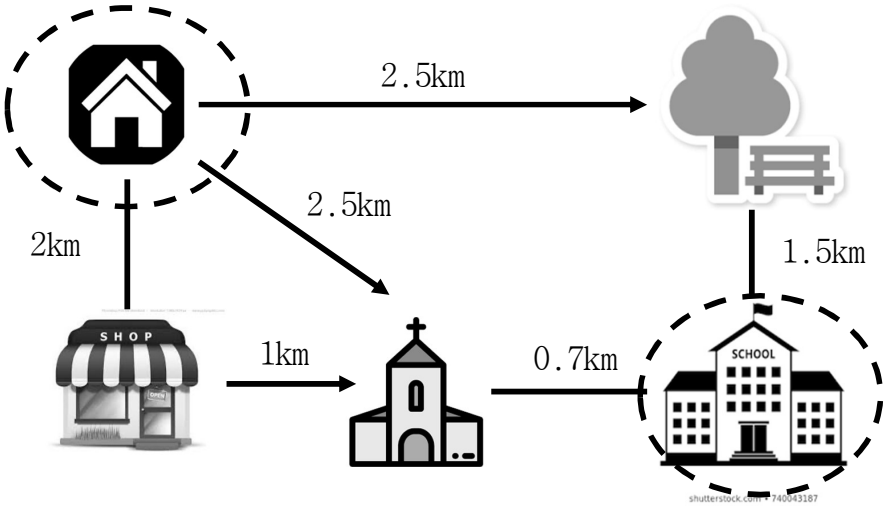
Hanyang University Introduction to Data Structures/J.S. Kim

Tree

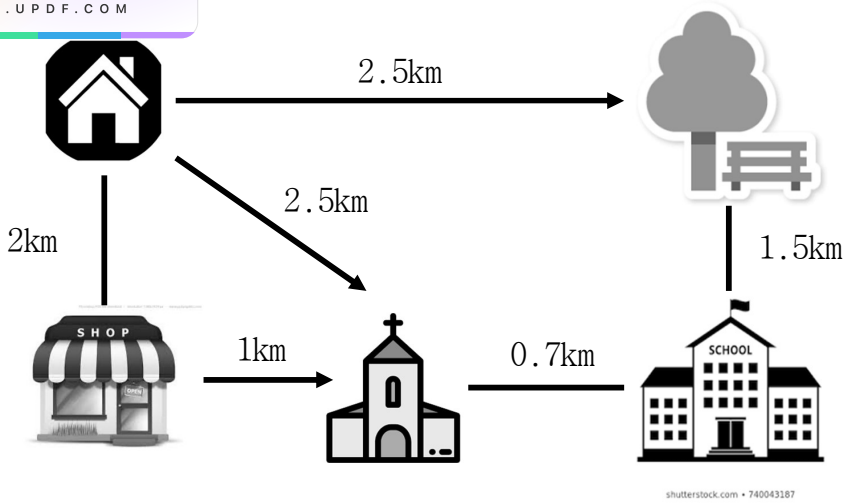


Graph

Why data structures are important?

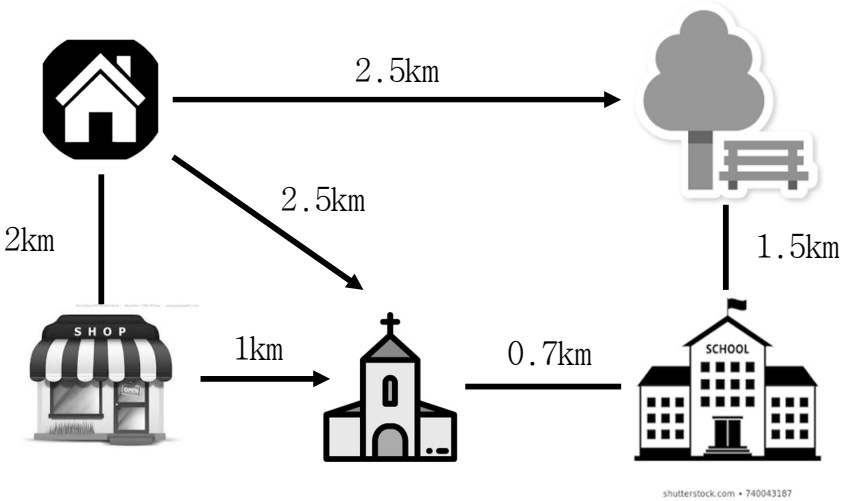


Home	**(100, 90)**
Shop	**(80, 90)**
Park	**(120, 110)**
Church	**(82, 100)**
School	**(78, 115)**



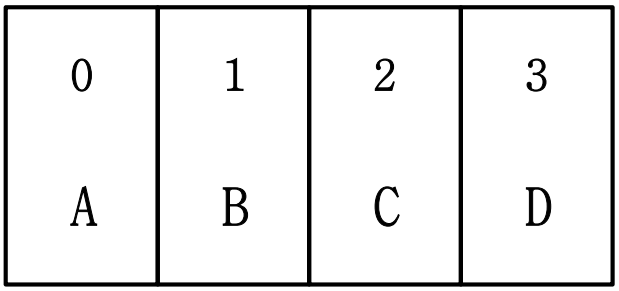
Problem:

- How to find the shortest path from home to school?



(Home, Shop, 2.0)
(Home, Church, 2.5)
(Home, Park, 2.5)
(Shop, Home, 2.0)
(Shop, Church, 1.0)
(Church, School, 0.7)
(Park, School, 1.5)
(School, Church, 0.7)
(School, Park, 1.5)

Home	**[(Shop, 2.0), (Church, 2.5), (Park, 2.5)]**
Shop	**[(Home, 2.0), (Church, 1.0)]**
Park	**[(School, 1.5)]**
Church	**[(School, 0.7)]**
School	**[(Church, 0.7), (Park, 1.5)]**



Locker[0] <= A

Locker[1] <= B

Locker[2] => C

Locker[3] => D

Locker = starting address

Locker[0] = Locker

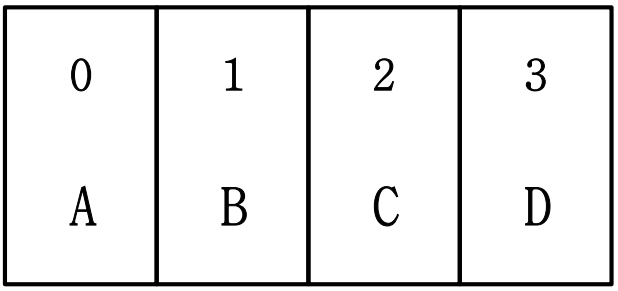
Locker[1] = Locker + 1 * sizeOf(item)

Locker[2] = Locker + 2 * sizeOf(item)

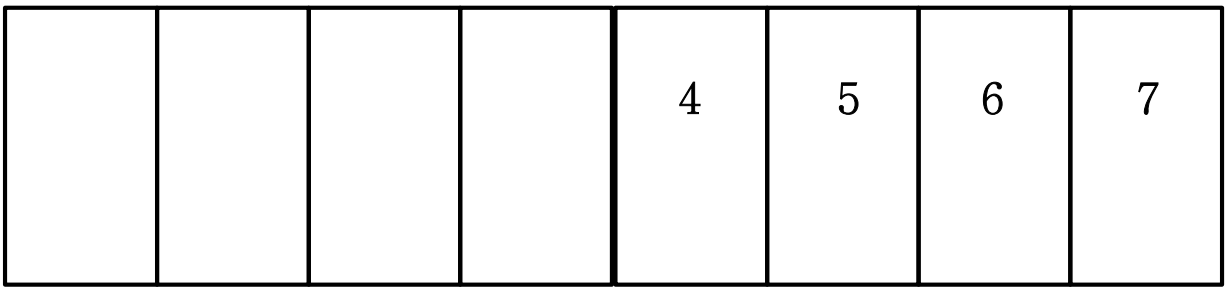
Locker[3] = Locker + 3 * sizeOf(item)

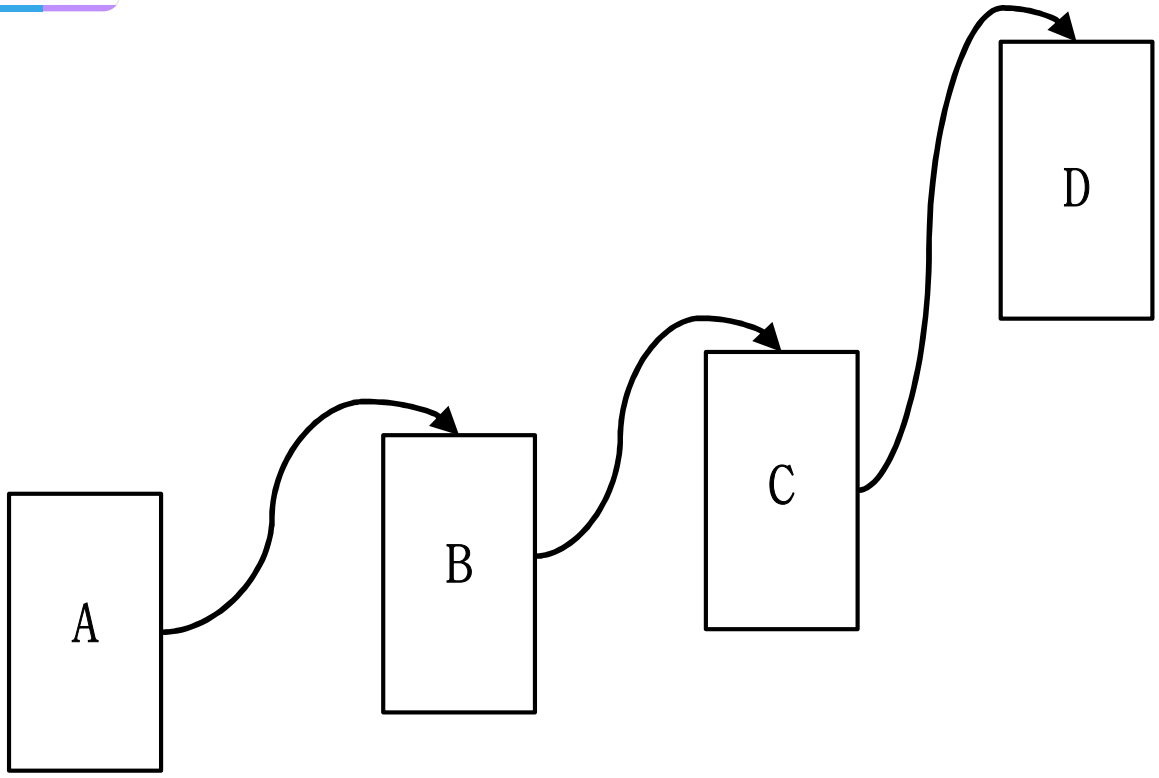
In C:

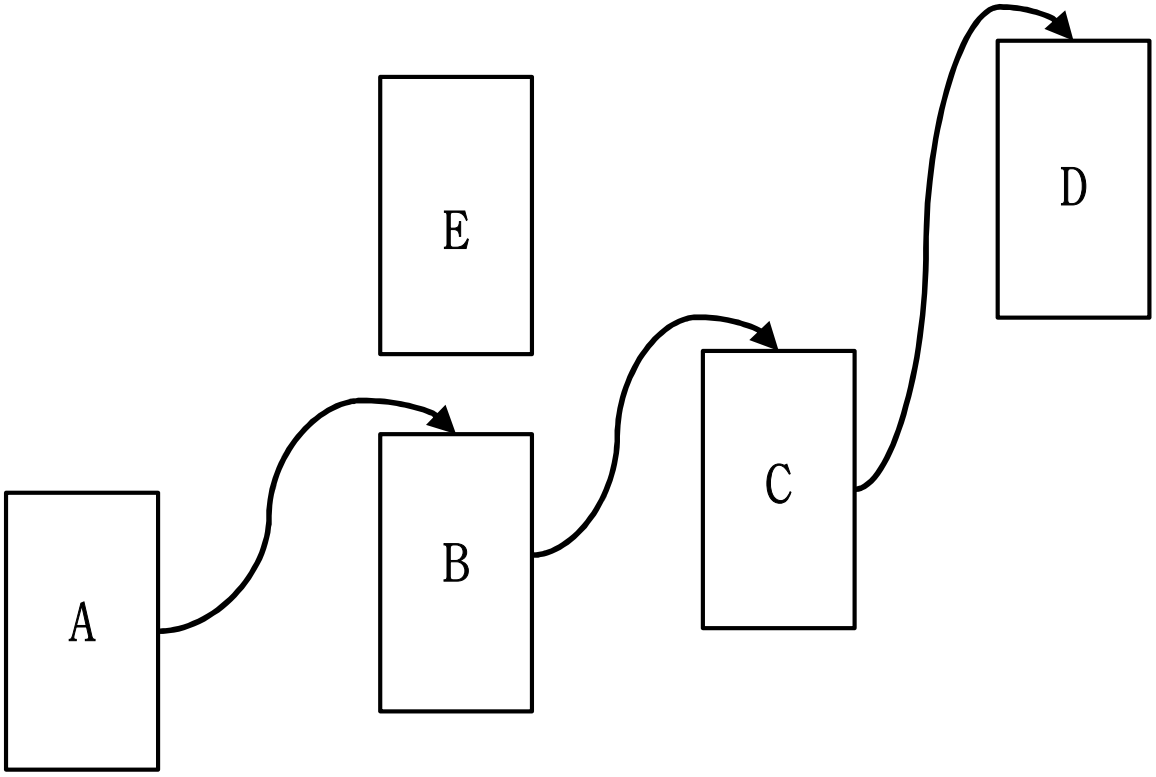
Locker[3] == *(Locker + 3)

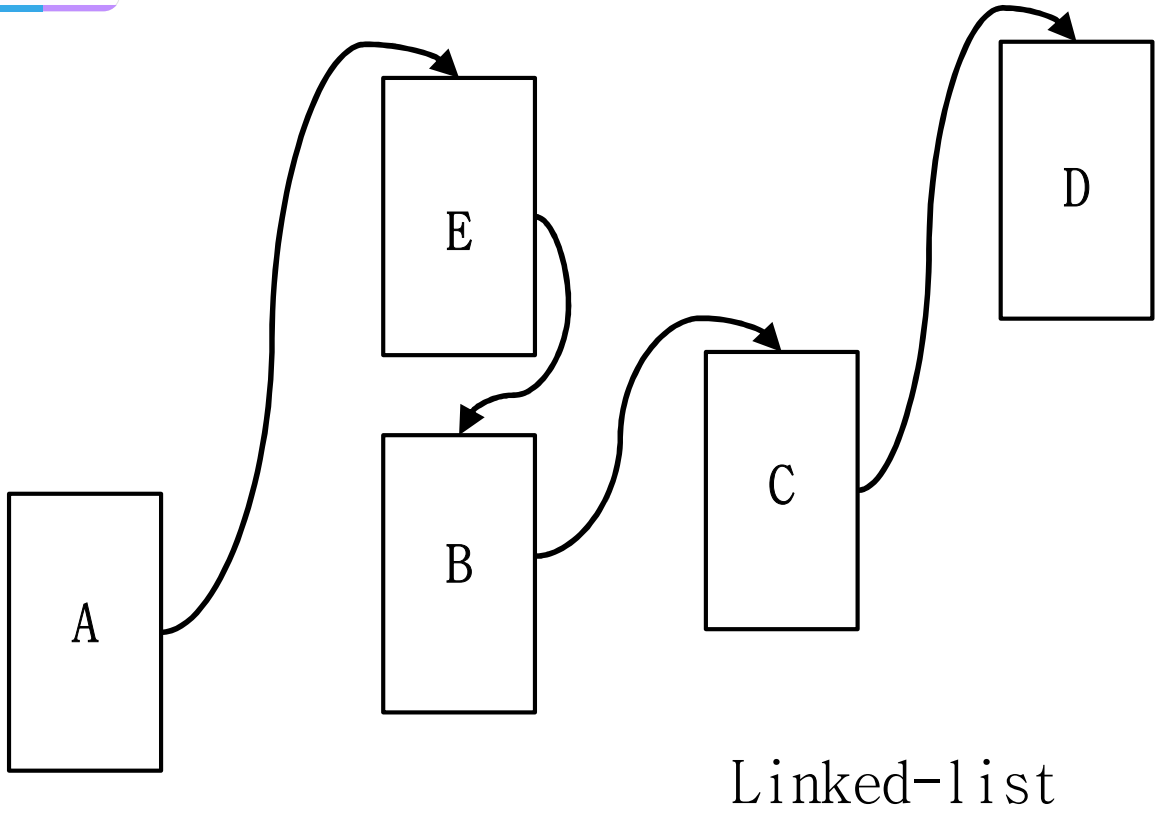


E

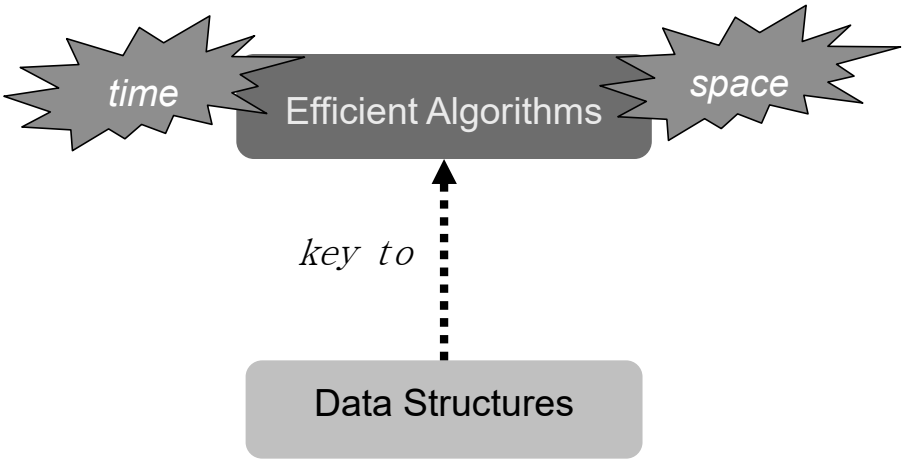








Data Structures and Algorithms



A well-designed data structure allows a variety of critical operations to be performed, using as few resources, both execution *time and memory space*, as possible.

Data Structures and Algorithms

Different kinds of data structures are suited to different kinds of applications.

Certain key tasks have algorithms that work best with particular data structures.

Relation database

Compiler

Social networks

Graphs

B+-Tree

Hashtables

Data Structures and Algorithms

Different kinds of data structures are suited to different kinds of applications.

Certain key tasks have algorithms that work best with particular data structures.

Relation database

Compiler

Social networks

Graphs

B+-Tree

Hashtables

Data Structures and Algorithms

Since data structures are so crucial, many of them are included in standard libraries of modern programming languages and environments, such as

C++'s Standard Template Library (STL) containers,

Java Collections Framework, and

Microsoft .NET Framework.

Therefore, you will learn ….

How to organize data?

- Commonly used data structures such as
  - Implementation dependent: *array*, *linked-list*
  - Implementation independent: *stack, queue, tree, heap, graph*, etc.
- Which operations are needed (Some more frequent than others)?

Algorithm analysis

- How do we compare two solutions?
- Theoretical study of algorithm performance and resource usage (Time complexity *vs.* space complexity)
- How will an algorithm behave (or scale) with the size of the input?

Algorithm design

- Sorting, searching, finding shortest paths, etc.

Q & A