DSA Revision

4009 DATA STRUCTURES & ALGORITHM (DSA)

Content Overview

Data Structure Trade-offs

Data structures	Advantages	Disadvantages
Array	Quick insertion, very fast access to index known	Slow search, slow deletion, fixed size
Ordered Array	Quicker Search than unsorted array	Slow insertion and deletion, fixed size
Stack	Provide Last-in First-out access	Slow access to other items
Queue	Provide First-in First out access	Slow access to other items
Linked List	Quick Insertion, Quick deletion	Slow search
Binary Tree	Quick search, insertion, deletion (if tree balanced)	Deletion algorithm is complex
Hash Table	Very fast access if key known, fast insertion	Slow deletion, access slow if key not known, inefficient memory usage
Heap	Fast insertion, deletion, access to large item	Slow access to other items
Graphs	Model real world situations	Some algorithms are slow and complex

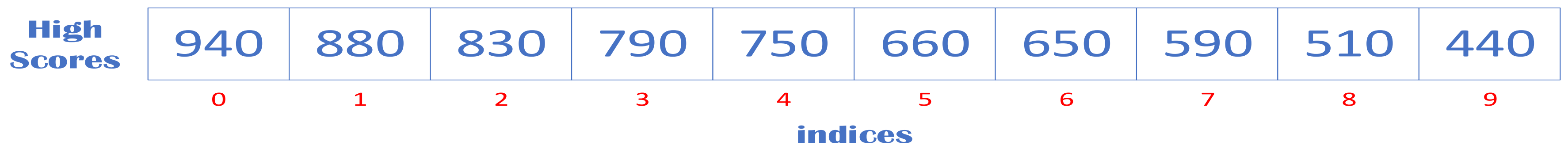
Week 5: Arrays

- Understand and implement Arrays in Java
- Able to create, access and understand Array elements and errors
- Discuss the out of bounds error in Java
- Understand and implement one or more of Array sorting algorithms
- Differentiate between Array sorting algorithms

Array concepts not in the exam

- multi-dimensional arrays

- The array is the most commonly used data storage structure; it’s built into most programming languages.
- Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.
- An array is a collection of items stored at contiguous memory locations. Therefore, an array is a numbered collection of variables all of the same



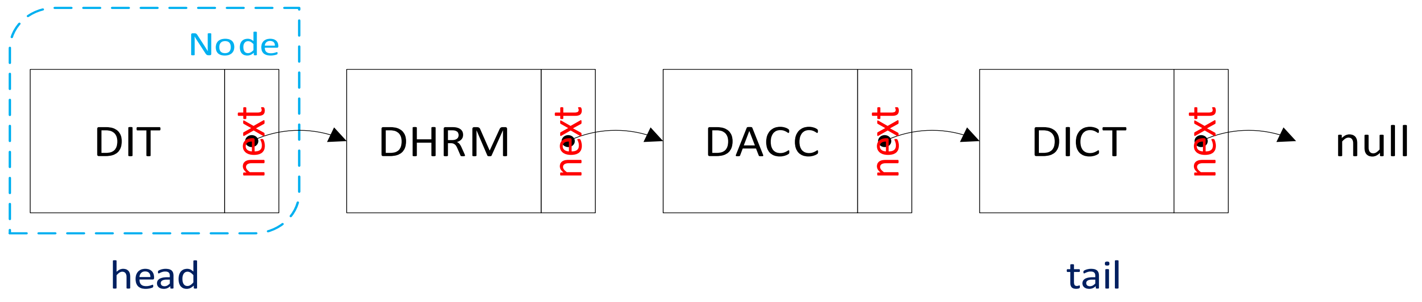
Week 7: Linked List

- ❖Understand and implement Linked Lists in Java
- ❖Identity and explain when linked list should be used instead of Arrays
- ❖Able to create, access and understand either Singly Linked List or Doubly Linked List
- ❖Differentiate between a Singly Linked List and a Doubly Linked List

- A linked list is a collection of nodes that together form a linear ordering. The ordering is determined as in the child's game "follow the leader" to which each node is an object that stores a reference to an element and a reference called next, to another node.

Linked List concepts not in the exam:

- RAM
- Variables & data structure in RAM



Week 12: Stack

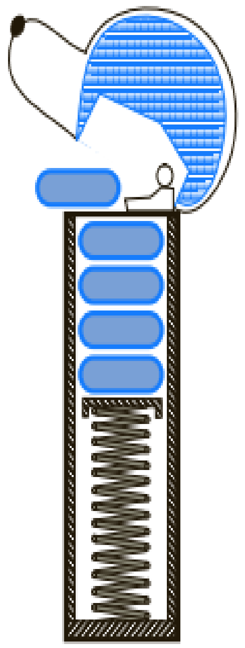
- ❖Stacks introduction
- ❖Application of stacks in the real world
- ❖Basic operations of stacks
- ❖Stacks (LIFO) implemented through Array and Linked List
- ❖A simple Array-Based Stack implementation

Stack concepts not in the exam:

- Generics

- Drawback of Array-Based Stack

- A stack is a collection of objects that are inserted and removed according to the last-in, first-out (LIFO) principle.



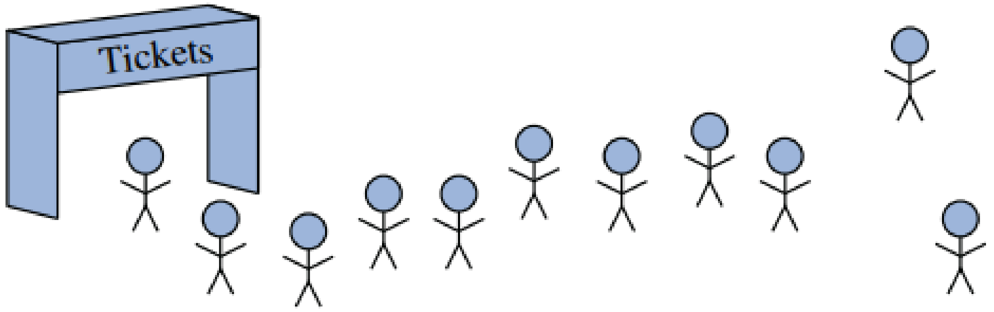
Week 13: Queue

- ❖Queues Introduction
- ❖The queue abstract data type
- ❖Example of a Queue data structure operation or method manipulation
- ❖Singly Linked List-based Queue implementation

Queue concepts not in the exam:

- Generics queues

- A queue is a collection of objects that are inserted and removed according to the first-in, first-out (FIFO) principle.



(a)

Week 14: Lists & Iterators

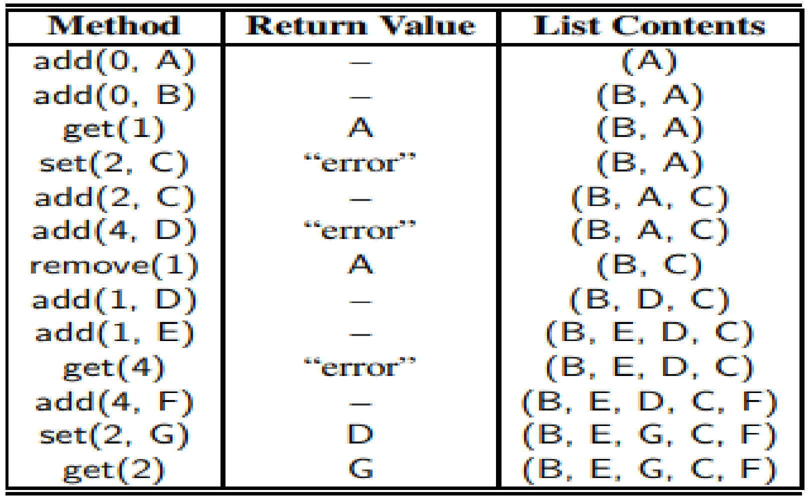
- The List ADT
- Application of Lists in the real world
- A simple version of the List Interface
- Different Lists in Java Programming
- Iterators
- Explanation of Iterator statement
- Explanation of the traditional (for loop method)
- Using the advanced looping method (For-Each loop)

Lists & iterators concepts not in the exam:

- Iterators

-Traditional and advanced looping methods

- The list is abstract datatype which represents a linear order sequence of elements. The list has support of adding and removing elements.
- The list data structure is efficient for data storage, data retrieval and where elements are accessed using an index position.



Method	Return Value	List Contents
add(0, A)	–	(A)
add(0, B)	–	(B, A)
get(1)	A	(B, A)
set(2, C)	"error"	(B, A)
add(2, C)	–	(B, A, C)
add(4, D)	"error"	(B, A, C)
remove(1)	A	(B, C)
add(1, D)	–	(B, D, C)
add(1, E)	–	(B, E, D, C)
get(4)	"error"	(B, E, D, C)
add(4, F)	–	(B, E, D, C, F)
set(2, G)	D	(B, E, G, C, F)
get(2)	G	(B, E, G, C, F)

Week 16: General trees & Binary trees

- General Trees
- Tree definitions and properties
- Formal Tree Definition
- General trees
- Simplified tree implemented data structure
- Binary trees
- The Binary ADT
- Simple binary tree implementation

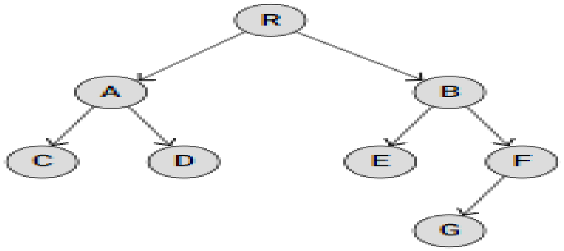
General trees & Binary tree concepts not in the exam:

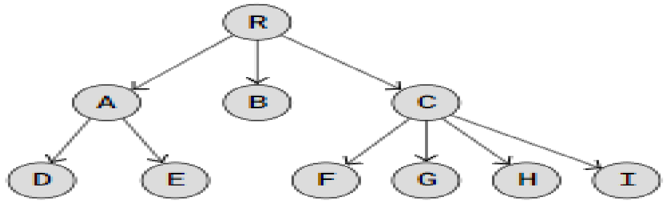
- Visual breakdown of tree ADT or binary ADT
- Coding of the tree or binary ADT

- Trees are non-linear data structure in computing.
- A tree is an abstract data type that stores elements hierarchically.

With the exception of the top element, each element in a tree has a parent element and zero or more children elements. Main terminologies for the tree data structure include parent, child, ancestor and descendant.

- A binary tree is an ordered tree with the following properties:
  - Every node has at most two children.
  - Each child node is labeled as being either a left child or a right child.
  - A left child precedes a right child in the order of children of a node.





Week 17: Map

- ❖Map ADT
- ❖Application of Map ADT in the real world
- ❖The list abstract data type (ADT)
- ❖Example of map operations
- ❖A simple map implementation
- ❖Student Directory App using Map data structure

Map concepts not in the exam:

- Map operations
- Advantage or disadvantages of Map ADT

- A map is an abstract data type designed to efficiently store and retrieve values based upon a uniquely identifying search key for each.
- A map stores key-value pairs (k,v), which we call entries, where k is the key and v is its corresponding value.
- Keys are required to be unique, so that the association of keys to values defines a mapping.

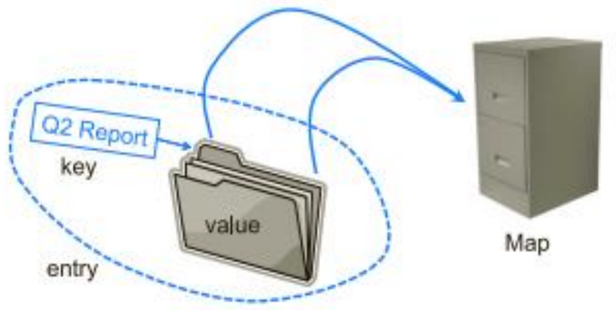


Figure 10.1: A conceptual illustration of the map ADT. Keys (labels) are assigned to values (folders) by a user. The resulting entries (labeled folders) are inserted into the map (file cabinet). The keys can be used later to retrieve or remove values.

Final Semester Exam Study Hints

Five part exam – 100 marks

Part A: Multiple choice questions (10 marks)

- 10 questions

Part B: Matching questions (10 marks)

- 10 questions – a data structure program with circled code statements will be given. Label the program according to the list given which explains the circled code statements.

Part C: Java Syntax questions (10 marks)

- 5 questions (discuss what the data structure method is performing or how to declare data structures)

Part D: Long Answer questions (25 marks)

- 6 questions (similar to weekly activity questions / differentiating & explaination questions)

Part E: Java programming (45 marks)

- 3 questions – code three whole Java programs with two method outputs

Study tips

- Revise the 7 lecture topics from week 5 to week 17.
- Make sure to complete week 5 to week 17 lab activities. Download the weekly activity solutions from the GitHub site
- Prepare for differentiating questions, use examples when possible to receive optimal marks
- Prepare to write the code for three Java Programs based on either Arrays, Linked List, Array List, Stack or Queue ADT. Each program ADT will require two method outputs. Weekly lab 1 provides clues

Exam schedule

Group	Venue	Time & Date
DIT D1	Seminar Room 3	8:30 – 10:30am – 19 June 2025
DIT D2	Seminar Room 8	8:30 – 10:30am – 19 June 2025
DIT D3	Seminar Room 8	8:30 – 10:30am – 19 June 2025
DIT D4	Computer Lab 3	8:30 – 10:30am – 19 June 2025
DIT E1	Seminar Room 8	8:30 – 10:30am – 19 June 2025