

Lecture 1.1 & 1.2

- Dr. Anurag Sharma

Data Structures

CS214, semester 2, 2018

1

What is a data structure?

- Data: are simply a value or set of values of different type which is called data types like string, integer, char etc.
- Structure: way of organizing information, so that it is easier to use.
- Data Structure: Its a way organizing data in such a way so that data can be easier to use.

CS214, semester 2, 2018

2

Organizing Data

- Any organization for a collection of records can be searched, processed in any order, or modified.
- Different kinds of data structures are suited to different kinds of applications
- The choice of data structure and algorithm can make the difference between a program running in a few seconds or many days.

CS214, semester 2, 2018

3

Which Data Structure or Algorithm is better?

- Must Meet Requirement
- High Performance
- Low RAM footprint
- Easy to implement

CS214, semester 2, 2018

4

Common Data Structures

- Array
- Stack
- Queue
- Linked List
- Tree
- Heap
- Hash Table
- Priority Queue

Data Types

A *data type* of a value (or object) is an attribute that tells what kind of data that value (or object) can have and a set of *operations* that act on those objects.

- For example, the data type *int* consists of the values *0, 1, 2*, etc. and the operations *+, -, *, /*, and *%*.

Data types in Java

- Primitive Data Types:
 - int, float, double, boolean, char
- Primitive wrapper classes (immutable):
 - Integer, Float, Double, Boolean
- Built-in classes:
 - Java has a rich set of built-in helper classes
- User Defined Data Types (UDT):
 - Use `class{...}` to create a UDT.

Data Structures with Java

- Arrays (single or multi-dimensional)
 - `String [] a = new String[SIZE];`
- Matlab does have any specific data types ☺. You can easily create an array (or matrix) just by specifying the values.
 - `A = [1 2 3; 4 5 6];`

Question

- Add two vectors in Matlab.
 - `A = [1 2 3 4];`
 - `B = [5 6 7 8];`
 - `C = A+B;`
- Add to vectors in Java.
 - Do it yourself 😊

(Doubly) Linked List and ArrayList (Dynamic Array)

- Java has a rich set of built-in data structures:
- Code to create a linked list:
 - `LinkedList<Integer> L = new LinkedList<Integer>();`
 - `L.add(10);`
 - `L.add(20);`
 - `L.remove(1);`
- Code to create a linked list:
 - `ArrayList<Integer> A = new ArrayList<Integer>();`
 - `A.add(10);`
 - `A.add(20);`
 - `A.remove(1);`

Homework

- What is the difference between Linkedlist and ArrayList?
- 2 different implementations of List class.
array list is dynamic array while LL is linked list**

Queue

- Queue data structure is based on First In First Out (FIFO)
- It has many applications where queuing is required such as message queuing.

Queue in Java ☺

- It is little tricky, however, implementation is not hard.
- You can simply use a LinkedList as Queue.
 - `Queue<Integer> q = new LinkedList<Integer>();`
 - `q.add(1);` //no specific enqueue method
 - `q.add(2);`
 - `q.remove();` //no specific dequeue method

Stack

- Stack data structure is based on Last In First Out (LIFO)
- Stacks have many applications from the parsing of algebraic expressions to keeping track of variables and return address values for function calls. (remember CS211?)

Stack in Java ☹

- No support! Try to use the built-in Stack<>. You will be disappointed.
- Solution?
 - Create your own stack ☺

```
public class MyStack <T> implements Iterable<T>{  
    private Stack<T> s;  
    private int maxSize;  
  
    @Override  
    public Iterator<T> iterator() {  
        return s.iterator();  
    }  
    ...  
}
```

Priority Queue

- Priority Queue is an extension of queue with following properties.
 - Every item has a priority associated with it.
 - An element with high priority is dequeued before an element with low priority.
- Applications: load balancing in OS, Dijkstra's algorithm, processing job requests based on priority.

Implementing priority queue

- `java.util.PriorityQueue` provides the basic operations of priority queue.
- ```
PriorityQueue<String> queue =
 new PriorityQueue<String>(SIZE, comparator);
```

  
where, 

```
Comparator<String> comparator =
 new UserDefinedComparator();
```

  
`//need understanding of inheritance to implement this.`

## Binary Tree in Java

- Create your own binary tree 😊
- Heap data structure
  - (Binary) heap is a complete (binary) tree (All levels are completely filled except possibly the last level).
  - The value of each node is greater (or less) than or equal to the value of its parent.

## Built-in algorithms for data structures

- You may use:
  - `Collections.sort(?)`
  - `Collections.min(?)`
  - `Collections.max(?)`
  - ...
  - `Arrays.binarySearch(?)`
  - `Arrays.sort(?)`
  - `Arrays.asList(?)`
  - ...