README.md

Session 8

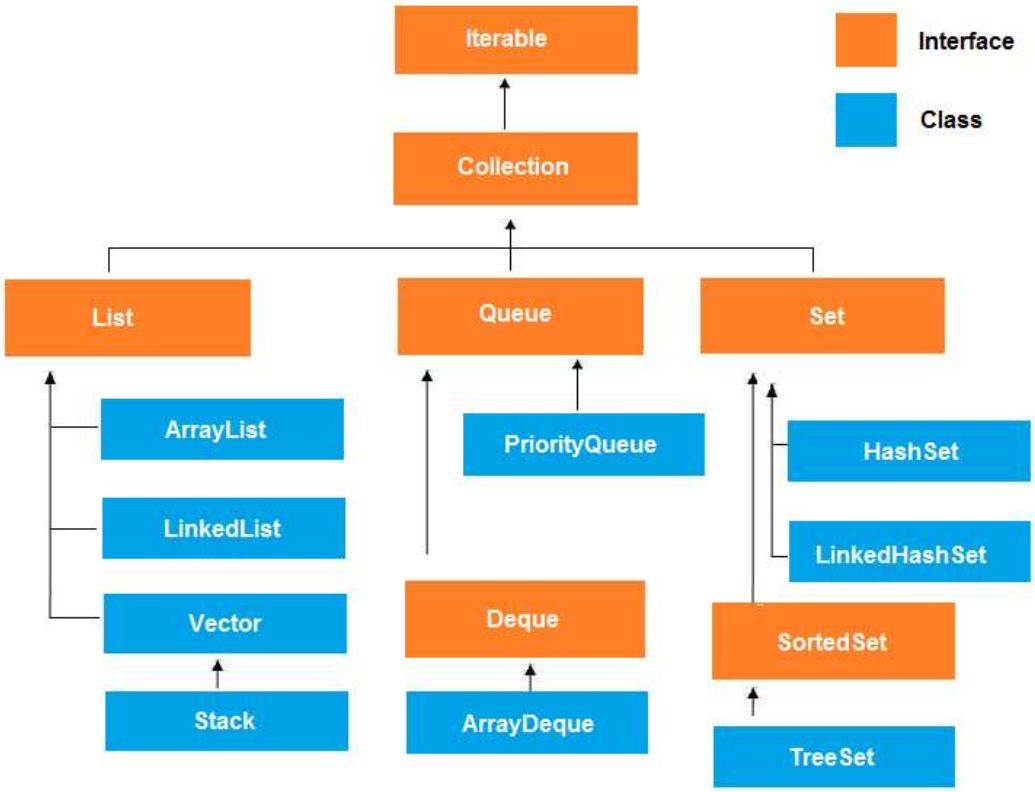
Table of Contents

The Collections Framework

The collection framework is a set of interfaces, implementations and algorithms, that allow you to store, retrieve, manipulate, and communicate aggregate data.

An excellent guide to the CFW is Oracle's Tutorial

The Collection Hierarchy (Part I)



Lists, Queues and Sets

For any collection, the most important resource is the JavaDoc. This will tell you everything you need to know about the class, right from constructors, methods and implementation details.

List: https://docs.oracle.com/javase/7/docs/api/java/util/List.html
Queue: https://docs.oracle.com/javase/7/docs/api/java/util/Queue.html
Set: http://docs.oracle.com/javase/7/docs/api/java/util/Set.html

Problems

1. You are given a long list of space separated words. There may be duplicates. Print the list of unique words, each one on a newline that appear in the previous list. It is **not** necessary to print them in the same order as they appeared in the list.
   Input:

   ```
   is it crazy how saying sentences backwards creates backwards sentences saying how crazy it is
   ```

   Output:

   ```
   is
   it
   crazy
   how
   saying
   sentences
   backwards
   creates
   ```

2. Can you modify the above code to print the words out in alphabetical order?

3. Given two sorted lists of integers of size n and m, create a new list that contains all the integers in sorted order.
   Input:

```
3 5
1
4
9
2
7
8
12
16
```

Output:

```
1
2
4
7
8
9
12
16
```

4. You are given a list of n numbers in no particular order. For each number, print out the next highest number in the list. If there is no such number, print -1.
   Input:

```
8
1 4 2 3 6 5 9 4
```

Output:

```
4 6 3 6 9 9 -1 -1
```

Explanation:

- For 1, the next highest number in the list is 4.
- For 4 the next highest number in the list is 6.
- For 2 the next highest number in the list is 3.
- For 3, the next highest number inthe list is 6.
- For 6 and 5 the next highest number in the list is 9.
- For 9, there is no next highest number.
- For 4, there is no next highest number.

5. You are given a list of n chess moves. Each move is a string, and can be either:

   i. Any string, such as "c5", "Be6", "Nf3", etc.
   ii. "undo".
   iii. "print"
   After each print command, print all the moves made till now in a space separated manner.
   In case the move is "undo", undo the last move. Input:

```
7
c5
print
Be6
Nf3
print
undo
```

```
print
```

Output:

```
c5
c5 Be6 Nf3
c5 Be6
```

Trees

Binary Trees

Binary Trees are a class of tree data structures, with the following property: *Each node may have a maximum of two children.*

Binary Trees by themselves are typically not very useful. By applying additional constraints on top of them, the can be given useful properties. Some examples of binary trees are:

1. Binary Search Trees
2. Heaps
3. Treaps

Problems:

1. https://www.hackerrank.com/challenges/tree-preorder-traversal
2. https://www.hackerrank.com/challenges/tree-postorder-traversal
3. https://www.hackerrank.com/challenges/tree-inorder-traversal
4. https://www.hackerrank.com/challenges/tree-height-of-a-binary-tree

Binary Search Trees

A Binary Search Tree (a.k.a BST) is a type of Binary Tree, which has the following constraints added to it:

1. The left subtree of the root node MUST have values less than the root node's value.
2. The right subtree of the root node MUST have values greater than the root node's value.
3. The left and right subtrees must be valid binary search trees.

A binary search tree gives you O(log(n)) access to any element, similar to using binary search in a sorted array.

Assignments

HackerRank

1. https://www.hackerrank.com/challenges/java-list
2. https://www.hackerrank.com/challenges/java-stack
3. https://www.hackerrank.com/challenges/java-hashset
4. https://www.hackerrank.com/challenges/tree-top-view
5. https://www.hackerrank.com/challenges/binary-search-tree-lowest-common-ancestor
6. https://www.hackerrank.com/challenges/binary-search-tree-insertion
7. https://www.hackerrank.com/challenges/is-binary-search-tree
8. https://www.hackerrank.com/challenges/maximum-element (solve using a CFW class)
9. https://www.hackerrank.com/challenges/simple-text-editor (solve using a CFW class)

Miscellaneous

1. In the directory binarytree/problems, there is an abstract class `BinaryTree`, and a concrete class `BinarySearchTree`. You need to fill in the missing methods, and *write a driver program to ensure that your implementation is correct*. A `TreeNode` class has already been provided for you.
   When reading the code, try to appreciate and understand how an abstract class is being used to provide a default implementation for some of the

methods. In particular, think about these points:

i. Why is `BinaryTree` abstract?

ii. Why are the methods `insert`, `delete`, `search` and `lowestCommonAncestor` abstract?