Lecture 2

Collections
“A group of things or people”

Last time:

We looked at the big picture in relation to choosing a data structure and its cost.

First, we briefly looked at what Big O Notation is and its major classifications.

- O(1): Constant
- O(log N): Logarithmic
- O(N): Linear
- O(N log N): Linearithmic
- O(N^2): Quadratic
- O(2^N): Exponential
- O(N!): Factorial

Second, we took a quick look at a few categories of data structures and major algorithms

- General-Purpose Data Structures: Arrays, Linked Lists, Binary Search Trees and Hash Tables.
- Special-Purpose Data Structures (ADTs): Stack, Queue and Priority Queue
- Sorting and Searching: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort, Linear Search and Binary Search
- Graphs

Good news is that Java provides a great framework for most of the data structures above, the Collections Framework.

In this lecture, we will have a simple and short overview of the Collections Framework.

Collection is:

> "A set of items or objects procured or gathered together by a person, group, or other agent." (source: wikitionary)

Collections are objects that group other objects.

1. You are a collection of ________________?
2. A dictionary is a collection of _________?

Also, collection is another term for data structures.

They are used to store, manipulate and retrieve aggregate data.

Collections in Java that are called the Java Collections Framework are a set of interfaces and classes that are available for you to use in the java.util package.

In relation to this course, there are three major interfaces in Java Collections Framework: List, Set and Map

Interface	Implementation
List	ArrayList, LinkedList
Set	HashSet, TreeSet
Map	HashMap, TreeMap, HashTable

In Java, the Collection interface is a group of objects, with duplicates allowed.

- Set extends Collection but forbids duplicates.
- List extends Collection also and allows duplicates and introduces positional indexing.
- Map extends neither Set nor Collection. Map is a collection of pairs (key, value). Map cannot contain duplicate keys.

Now, who, as application programmers, are we?

We are users of the Collections. The first thing we need to care about, as a user of the Collections, is what kinds of operations we can perform on them.

The common operations are:

- Addition (Insertion)
- Removal (Deletion)
- Sorting
- Searching
- Iteration (Traversal)
- Copying (Cloning)

It sounds like it is not necessary for us to care about how the Collections, or the data structures, are implemented. Hooray!

But, aren't you curious? Are you ready to explore?