Sorting and Searching -- Introduction

- Two common programming tasks are sorting a items and searching for an item in a list
- Chapter 13 focuses on:
  - Selection Sort
  - Insertion Sort
  - A generic sort for objects
  - Linear Search
  - Binary Search
  - Hashing

Sorting

- Sorting is the process of arranging a list particular order
- There must be some value on which the order
- There are many algorithms for sorting a lis
- These algorithms vary in efficiency
- We will examine two specific algorithms: S and Insertion Sort

Selection Sort

- The approach of Selection Sort:
  - select one value and put it in its final place ir
  - repeat for all other values
  -
- An expanded version:
  - scan the list to find the smallest value
  - put it in the first position
  - find the next smallest value
  - put it in the second position
  - repeat until all values are placed

Selection Sort

- An example:

original:	3	9	6	1	2
smallest is 1:	1	9	6	3	2
smallest is 2:	1	2	6	3	9
smallest is 3:	1	2	3	6	9
smallest is 6:	1	2	3	6	9

- See Selection_Sort_Test.java

Insertion Sort

- The approach of Insertion Sort:
  - Pick any item and insert it into its proper place
  - repeat until all items have been inserted
  -
- An expanded version:
  - consider the first item to be a sorted sublist (o
  - insert the second item into the sorted sublist, s necessary to make room to insert the new addition
  - insert the third item into the sorted sublist (of shifting as necessary
  - repeat until all values are inserted into their p

–

Insertion Sort

- An example:

original:	3	9	6	1	2
insert 9:	3	9	6	1	2
insert 6:	3	6	9	1	2
insert 1:	1	3	6	9	2
insert 2:	1	2	3	6	9

-
- See Insertion_Sort_Test.java

Comparing Sorts

- Both Selection and Insertion Sorts are simi
  efficiency
- The both have outer loops that scan all ele
  inner loops that compare the value of the c
  almost all values in the list
- That is approximately n^2 number of comparis
  list of size n
- We therefore say that these sorts are of or
- Other sorts are more efficient, on the order of $n \log_2 n$

Sorting Objects

- Integers have an inherent order
- But the order of a set of objects must be d person defining the class
- Recall that a Java interface can be used as and guarantees that a particular class has particular methods
- We can use this to develop a generic sort f objects
- See Object_Sort_Test.java

Searching

- The goal is to find a particular target val
- In an unsorted list, a linear search is a simple technic
- Scan through the list one item at a time in for the target value
- See Linear_Search_Test.java
- If the list is sorted, not all items need t
- As soon as you encounter a value greater th value, you know the target is not in the li

Binary Search

- If the list is sorted, a binary search can be used to find a target value
- Strategy: examine the item in the middle of the list, eliminating half of the items from consideration
- The target value will either be found, or will be in one half or the other
- Continue the process, examining the middle item of the search section on each comparison
- See Binary_Search_Test.java

Recursive Binary Search

- The binary search algorithm can be implemer recursively
- It has the same general strategy as the ite
- Each recursive call narrows the search sect
- See Binary_Search_Test2.java

Hashing

- Hashing is another technique for storing an information
- A hash function produces a hash code for any given item
- The hash code is scaled to a hash index rel size of the hash table
- The item is stored in hash table at the has

Hashing

- Collisions may occur, resulting in a list o particular index
- Therefore a small search may be necessary
- Because hashing is based on calculations, i efficient
- Storing and searching involve the same amou