Heap

1. Basic Concepts
   - A heap is a complete binary tree used to store elements that have an ordering
     - Each node stores an element
     - The element in a node must be less than elements in children nodes

2. Heap Implementation
3. Applications of Heap

Tasks of the Week

- Work through the insertion and deletion operations on an array implementation of a heap.
- Case Study: ArrayListHeap - ArrayList implementation of HeapADT
- Case Study: Implementing a priority queue with a heap
- Case Study: The heap sort
public interface HeapADT<T extends Comparable> {
    public void addElement(T obj);
    public T removeMin();
    public T findMin();
    public int size();
    public boolean isEmpty();
}
Add Element

- Always add new elements to the end of the tree. But this may result in a violation of the heap property.
- To restore the heap property, move towards the root from the newly added node, at each node,
  - if the element in the node is larger than the element in the parent node, switch the two elements and move to the parent node
  - Otherwise, stop.
- Time complexity is $O(\log_2 n)$ for a heap of $n$ nodes.

Remove Minimum Element

- Remove the element in the root and fill the root with the last element (also remove the last node).
  - This will lead to a violation of the heap property.
- Restore the heap property top-down starting at the root. At each node,
  - If the element in the node is not the smallest compared to elements in child nodes, switch the elements in this node and in the child node that contains the smallest element. Continue the restoration at that child node.
  - Stop if a leaf node is reached or no exchange is possible.
- Time complexity is $O(\log_2 n)$ for a heap of $n$ nodes.
Implement PriorityQueue Using Heap

- A HeapPriorityQueue class can be implemented as a wrapper of an ArrayListHeap class
  - Every PriorityQueueADT method can be implemented by a method of a Heap class
- Alternatively, a HeapPriorityQueue class can be implemented as a subclass of a Heap class, say the ArrayListHeap class

Heap Sort

- To sort a list of elements in ascending order,
  - Add the elements into a heap
  - Then remove the minimum element of the heap and append it to the output list, until the heap is empty
- Time Complexity is $O(n \log_2 n)$ for a list of $n$ elements