**CSE 274**

**Heap and Graphs**

Write the following methods:

// **(15 pts)** Return true if the array is a legal heap. Returns false if all other situations.

**public static <T extends Comparable<T>> boolean isHeap(T [] heap, int heapSz)**

// **(15 pts)** (See exercise 26.5) Change an element in the heap. This change will usually violate the heap property.

// Efficiently make the change, while making any necessary repairs.

**public static <T extends Comparable<T>> void changeHeap(T [] heap, int heapSz,  
 int whichElement, T newValue)**

// **(70 pts)** (See project 28.7) Write method to determine if an NxN N-Puzzle is solvable. The class

// will have the following definition:  
  
**public class NPuzzle {**

// Determine if a matrix representing an NxN puzzle is solvable.  
 // A legal board will be square and contain values 1, 2, ... N2. The empty square will be represented

// with a -1.Throws exception is board is, in way, nonsensical.  
  **public static boolean solvable(int [][] board)**  
**}**  
  
Additional methods and classes can be added to this definition.  
  
Here are two simple tests for you conduct:  
int [][] v1 = { {-1, 1, 2},  
 { 5, 6, 3},  
 { 4, 7, 8}};  
int [][] v2 = { {-1, 2, 1},  
 { 5, 6, 3},  
 { 4, 7, 8}};  
System.out.println(NPuzzle.solvable(v1)); // Should be true  
System.out.println(NPuzzle.solvable(v2)); // Should be false  
  
You may use standard Java set/map classes. If you do, they must be HashSet and HashMap. The approach that you use must be based on the approach used in the Nim example. You may use Java Queue classes too.