**CSE 274**

**Big O and Linked Data**

1. **(60 pts)** The following starter code is provided to you (located in homework folder). ResizableSet and LinkedSet are to be ADTs that implement the interface SetInterface. ResizableSet is done for you and is based, almost in whole, on the textbook’s resizable array bag.  
     
    SetInterface.java ResizableSet.java LinkedSet.java  
   1. **(15 pts)** ResizableSet is an implementation of a set using a resizable array. Determine the best-case and worst-case Big O for the following methods. So, for each method listed, you will have 2 answers - best-case and worst-case. Assume N is the number of elements in the set. FYI: the length of the array is, at most 2N. Therefore, the length of the array is O(N).  
      1. getCurrentSize
      2. isEmpty
      3. add
      4. remove
      5. clear
      6. contains
      7. toArray
      8. getCapacity
   2. **(45 pts)** Implement the ADT set (named LinkedSet) using doubly linked list to contain the entries. This class must implement the interface SetInterface. The linked list must be a doubly linked list. When the TA tests your code, he will insert code to ensure that the prev and next links are properly set.  
        
      You need to provide code that shows you thoroughly tested your classes.  
        
      There are notes in the starter code that list other required functionality.
2. **(9 pts)** Complete the following Java function that utilize the standard Java Stack class. Test your code.  
    /\*\*  
    \* Rotates the top n elements of the stack. That is, when n = 3 and the stack  
    \* looks like (vertical bar shows bottom of stack:  
    \* | A B **C D E**  
    \*   
    \* The top element is moved from the top and put into the nth position.  
    \* For the above example, the stack becomes:  
    \* | A B **E C D**  
    \*   
    \* @param stk The stack to be processed  
    \* @param n   
    \*/  
    public static <T> void rotate(Stack<T> stk, int n) {  
    }
3. **(15 pts)** At the end of Chapter 4 is a section named “Projects.” A short piece of code is shown to show how to time Java source code. You will need this code to complete the following project. Present your timings in a graph; Excel is a good way to present data.
   1. Do Project #2 (N vs N2 algorithm)
4. **(4 pts)** Determine the best case and worst case Big O for the following piece of code that processes a NxN 2D array.  
   1. for (int r=0; r<N; r++)  
       for (int c=r; c<N; c++)  
       if (ary[r][c] == key)  
       return true;  
      return false;
5. **(12 pts)** Determine the worst case Big O for the following pieces of code. Assume that img is an NxN array.
   1. for (int r=0; r<N/2; r++)  
       for (int c=r; c<N/2; c++)  
       sum += img[r][c];
   2. for (int i=3; i<N-3; i++)  
       sum += i \* j;
   3. string str = "A";  
      for (int i=0; i<N; i++)  
       str += str;  
      for (int j=0; j<str.length(); j++)  
       sum += str[j];
   4. void f(int N)  
      { return 2 \* N; }  
      for (int i=0; i<N; i++) {  
       sum += f(i-1);  
       sum += f(i);  
       sum += f(i+1);  
      }
   5. for (int r=0; r<N; r++)  
       for (int c=0; c<N; c++)  
       for (int dr=-1; dr<=+1; dr++)  
       for (int dc=-1; dc<=+1; dc++)  
       res[r][c] += img[r+dr][c+dc];
   6. for (int r1=0; r1<N; r1++)  
       for (int c1=0; c1<N; c1++)  
       for (int r2=0; r2<N; r2++)  
       for (int c2=0; c2<N; c2++)  
       sum += img[r1][c1]\*img[r2][c2];