Lab 13: Longest Increasing Subsequence

Professor: Ronaldo Menezes

TA: Ivan Bogun

Department of Computer Science
Florida Institute of Technology

November 12, 2013

1 Problem statement

1.1 Legend

Longest increasing subsequence is the problem where given an array of elements find the longest sorted subsequence [1]. You are asked to solve it using :

- recursion
- dynamic programming

2 Implementation

Implement the class *LongestIncreasingSequence.java*

```
// LongestIncreasingSequence.java
public class LongestIncreasingSequence {

    private int[] sequence;

    public int lisRecursive(){
    // implement recursive solution to the longest increasing subsequence, only
        the length of it should be returned.
    }

    public int lisDynamic(){
    // implement dynamic programming solution to the longest increasing
        subsequence, only the length of it should be returned
    }

    public int[] getLongestIncreasingSubsequence(){
    // return array which is the longest increasing subsequence of the
        'sequence'. Note that such array is not unique, however the length of
        such array is unique.
    }

    }
```

3 Extra-credit(+10)

Solve the problem in $O(n \log n)$ time. Name the function *lisFast()*.

[1] http://en.wikipedia.org/wiki/Longest_increasing_subsequence

4 Sample input-output

4.1 Input

Use the following main for testing.

```
	public static void main(String[] args){
		int[] a={2,1,4,5,7,2,3,1,2,3,8};

		LongestIncreasingSequence lis=new LongestIncreasingSequence(a);

		int recursiveSolution=lis.lisRecursive();
		int dynamicProgrammingSolution=lis.lisDynamic();

		int[] array=lis.getLongestIncreasingSubsequence();

		System.out.println("Recursive solution:    " +recursiveSolution);
		System.out.println("Solution via dynamic programming:
		    "+dynamicProgrammingSolution);

		System.out.println("Longest increasing subsequence itself: ");
		for (int i = 0; i < array.length; i++) {
			System.out.print(array[i]+" ");
		}

	}
```

4.2 Output

```
Recursive solution:        5
Solution via dynamic programming:  5
Longest increasing subsequence itself:
2 4 5 7 8
```

5 Grade breakdown

basis	grade
Implementation	(60)
dynamic programming solution	20
recursive solution	20
sequence itself	20
Comments	(20)
Javadocs	10
General	10
Overall	(20)
Compiled	5
Style	5
Runtime	10
Total	100(+10)