Data Structures

Recursion Basics and mergesort Lab

In this lab we will experiment with the

1. basic principles of recursion
2. Discuss the idea of memoization –a technique that uses memory space to improve speed of execution by storing previously calculated values.

**Part 1:**

Each of the following functions will be coded.

* For each you must identify the base case(s)
* Code the recursive step.

1) Triangular Numbers

* Triangle(1) =1
* Triangle (n) = n + Triangle(n-1)

2) Factorial:

* 0 factorial =1
* 1 factorial =1
* n!= n \* (n-1)!

3) Fibonacci Series

* + Fib(0)=0
  + Fib(1) =1
  + Fib (n)= fib(n-1) + fib(n-2)

4) Create a recursive function ***printStars***(***int n***) that prints a triangle of stars of size n.

For example if n=4

\*\*\*\*

\*\*\*

\*\*

\*

Should be printed**Part 2**

Merge Sort

Objective: Complete a class that implements an Unordered array (Outlined below).

|  |
| --- |
| OrderedArray |
| m\_array (type int[])  maxSize (type int)  numElements (type int) |
| Constructor (int size)  addLast (int item)  removeLast ()  efficientRemove(int index)  linearSeach(int item)  listItems ()  insertionSortAsc()  selectionSortDesc()  **mergeSortAsc()** |

Main functions defined below:

**In this Lab**

**1) The mergesort function will be added to the previous solution from lab 1 to 3 (unordered array) posted on blackboard.**