Ipsit Patra (UIN: 673438945) | CS401 | April 15, 2014

Dynamic Programming Project

Synthesis of trees of gates

# Summary

A *working* Java program of Synthesis of trees of Gates i.e. construct a legal tree of gates which minimizes the arrival time at the output of the tree, implemented using dynamic Programming. The Program has no limitations and will work as long as the conventions are followed.

## Formulation 1:

Given: A sequence of n arrival times t0, t1,..., t(n-1), and a gate delay dg (as a float).

Goal: the time is minimized, regardless of cost.

# Division of Labor

The project was completed individually and there are no team members involved.

# Algorithm Description

1. Create an array of input arrival time using tree data structure acting as leafs.
2. Create an array of cost and delay and extract the gate with the minimum delay time in the library file.
3. Find the minimum two arrival times of the input arrival pins timings considering consecutive subsequence of length two. The gate will be implemented using tg=max(tx,ty) + dg where tg is the final gate delay and dg delay of the selected gate from step 2.
4. The gate will act as parent for the two leafs from step 3.
5. Remove the two arrival times from the array and add the new element at the index that the two arrival times are removed.
6. Repeat the steps from 3 to 7 until the array contains just one element.
7. The element is the root element for the tree.

## Runtime

If N is the number of arrival input pins and I is the number of gates, then the runtime of the algorithm would be O(N+I) as delete, insert and sorting on an array of N elements takes O(N) time.

# Solution Extraction

Only basic operations on the arrayList are performed and the recursive function use the maximum function and do the operations accordingly.

# Details

Memoization is not used but Recursive Function and Tree Data Structure are used

# Example Instance

### Input File

6

6.0

3.0

4.0

7.0

2.0

3.0

### Library File

7

1.1 8

3.2 3

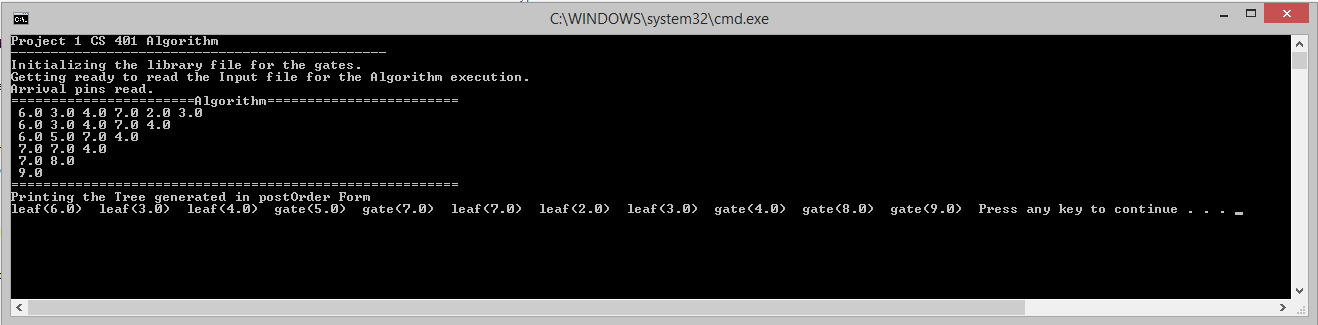
3.4 6

3.7 9

1 4

7.9 1

3.5 6



### Output

### Project 1 CS 401 Algorithm

### -----------------------------------------------

### Initializing the library file for the gates.

### Getting ready to read the Input file for the Algorithm execution.

### Arrival pins read.

### =======================Algorithm========================

### 6.0 3.0 4.0 7.0 2.0 3.0

### 6.0 3.0 4.0 7.0 4.0

### 6.0 5.0 7.0 4.0

### 7.0 7.0 4.0

### 7.0 8.0

### 9.0

### ========================================================

### Printing the Tree generated in postOrder Form

leaf(6.0) leaf(3.0) leaf(4.0) gate(5.0) gate(7.0) leaf(7.0) leaf(2.0) leaf(3.0) gate(4.0) gate(8.0) gate(9.0)

Press any key to continue . . .