COSCI 21a, Assignment 4

Directions: To receive full credit:

• Place your name at the top of each page.
• Start each problem on a new page.

1. A complete binary tree is one where every non-leaf vertex has exactly two children and every leaf has the same depth. Prove by induction that for a complete binary tree of height \( h \) (height 0 is a single vertex, height 1 is a vertex with two children, etc.), the number of vertices is:

\[ 2^{h+1} - 1 \]

2. Give a non-recursive algorithm that performs pre-order traversal of an arbitrary (unordered) tree of \( n \) vertices by employing a stack. Assume that you have available the stack operations initializing an empty stack, testing if the stack is empty, PUSH, POP, and TOP (where TOP returns the top element but does not remove it). Assume also that an array representation of stacks is being used.

3. Define the child number of a tree to be the maximum number of children that any vertex has. Let \( T \) be an unordered tree stored with the LMCHILD-RSIB representation. Describe in English and give pseudo-code (using the PARENT, LMCHILD, and RSIB functions) for a recursive algorithm to compute the child number of \( T \). Analyze the asymptotic time and space used.

4. For a vertex \( v \) in a binary tree, let LCHILD(\( v \)) denote the left child of \( v \) (or \( nil \) if \( v \) does not have a left child) and RCHILD(\( v \)) denote the right child of \( v \) (or \( nil \) if \( v \) does not have a right child). Give a proof by induction that the following algorithm computes the height of the subtree rooted at \( v \) in a binary tree (or returns \(-1\) if \( v \) is \( nil \)).

function BHEIGHT(\( v \)):
   if \( v=\text{nil} \) then return \(-1\)
   else return \( 1+\text{MAXIMUM}\{\text{BHEIGHT}(\text{LCHILD}(v)),\text{BHEIGHT}(\text{RCHILD}(v))\} \)
end