CMSC 420
Programming Assignment 1
Tries

Your first project involves implementing insertion, deletion and lookup algorithms for compressed Patricia tries. However, unlike the data structure that we went over in class, your data structure will not encode the substring information with the edges of the trie. Each node of your trie will be a record containing **only** the following information:

1) The index of the character in the input string to test at that node.
2) 3 pointer fields – one for each of the 2 possible next bit values (0 or 1) in the string, and a 3rd for the end of string symbol. If the pointer in the end of string position is non-null, it points to a copy of the string that ends at that position.

Except for the root, the compressed trie does not contain any node that has only a single, non-null pointer unless that pointer is the end of string pointer. So, if u and v are two nodes in the compressed trie with u the father of v, I_u the index of u, I_v the index of v, and c_{uv} the bit whose pointer takes you from u to v, then all you know is that:

$$\text{prefix}(v) = \text{prefix}(u) + c_{uv} + \ ?$$

where ? is an unknown bit string of length $I_v - I_u - 1$. To determine what bit string ? corresponds to, you would have to search the sub-trie rooted at v for a node containing a non-null end of string pointer, retrieve the bit string and look up the bits in the $I_u + 1^{st}$ position to the I_v^{th} position. Your insertion algorithm can only conduct this search after it has determined that the string to be inserted is not already in the trie – i.e., you CANNOT attempt to determine what ? is every time you move along an edge.

The assignment will be graded as follows:

1. Definition of algorithms and data structures. You will define the record structure of the compressed trie in C++ and provide a narrative (i.e., English!) description of your algorithms for lookup, insertion and deletion. The algorithms should be illustrated with an example trie to which you apply the operations. This part of the assignment is worth 40% of your grade, and if I can't read/understand what you submit your grade will reflect that. Your submission here is limited to 4 pages, max!
2. Submission of code and analysis of correctness of implementation. Your code will be tested by a sequence of insertions, deletions and lookups into an initially empty trie. After each operation you will output a level order traversal of the trie, which will be used to grade your project. For each node in the level order traversal you output (index of father in traversal, bit in father that takes you to current node, level of current node, string associated with node if this node has end of string pointer set (null string otherwise)). The TA will provide details on the submission process and formats of input and output data.

The assignment is due by midnight October 15. The TA will post late penalties.