COSC329 Tutorials on Combinatorial Generation

(1) Make an algorithm for generating binary strings in reverse lexicographic order.

(2) Make an algorithm for generating ternary strings in lexicographic order.

(3) Draw the lexicographic tree of ternary strings of length 3. There are 27 leaves corresponding to the strings. Two strings 212 and 213 share the same path 21, etc.

(4) Make a recursive algorithm for generating ternary strings with one change from string to string. Hint: Follow the algorithm for the binary reflected Gray code.

(5) Make the twisted lexicotree for the above set of strings.

(6) Convert the above algorithm into an iterative one.

(7) Make an algorithm for generating permutations in reverse lexicographic order.

(8) In Johnson's algorithm, n moves most frequently, n-1 moves next most frequently, ..., etc. Convert the algorithm so that 1 moves most frequently, 2 moves next most frequently, ..., etc.

(9) Prove that the last permutation by Johnson's algorithm is 2 1 3 4 ... n

(10) A meta program is one that generate a nested loop structure such as Algorithm 9. Make such a meta program and run it. As a result you have a generated program. Run it and confirm you can generate combinations with an output statement. Compare the time with the recursive one for a large n and r. Remove the output statement for a large experiment..

(11) Make a recursive algorithm that generates parenthesis strings in reverse lexicographic order.

(12) Draw all binary trees with 4 internal nodes and confirm the number is 14. Give the correspondence between the parenthesis strings and the trees.

(13) Draw all the diagonals of a polygon with (n+2) vertices. The number is the n-th Catalan number. Confirm this with n=4. Give the correspondence between parenthesis strings and the diagonals.

Example with n=3. Originate a parenthesis string from 0. If you do not draw diagonal from 0 any more, go to 1, etc. In the second figure, we embrace a polygon of n=4.