

CS5403 Data Structures and Algorithms

Assignment 4

Assigned date: October 8, Thursday

Due date: October 22, Thursday

Do not use a separate header file, application file and implementation file. Instead, put all your code in a single file called hw4.cpp and submit it via the NYU Classes site.

Please include the return 0; statement in main() in this assignment and all future assignments.

Tasks:

- Write a recursive function, which will generate all mathematically valid paranthesization schemes for a given number of pair of parenthesis. For instance for 1 pair of parenthesis, the only valid paranthesization is:

()

For 2 pairs of parenthesis, the valid paranthesizations are:

(())
() ()

Notice that “ ()) (“ is NOT a valid paranthesization. So essentially in a valid paranthesization, all open paranhteses should be closed.

For 3 pairs of parantheses, the valid paranthesizations are:

((()))
(() ())
() (())
() () ()
(()) ()

You should notice a recursive pattern emerging. In this assignment you are supposed to implement this recursion. Your task is to write a recursive function which takes number of parenthesis pairs as input and recursively generates all valid paranthesization schemes for that given number of parenthesis pairs.

- Your other task is to write a main function, which interacts with the user. Basically your main function should ask the user for the number of parenthesis pairs. After the user provides the number, your program should generate and print out all valid paranthesizations.

Hint: You may be generating duplicate paranthesization schemes at various points during your recursion. You may use STL's `unordered_set` data structure to deal with this. Basically, as the name suggests, `unordered_set` is a set and it ignores duplicate values. So, as you generate valid paranthesizations, you can store them in `unordered_set`. And then when you are done, you can iterate over the set and print out its contents. For more details see

http://www.cplusplus.com/reference/unordered_set/unordered_set/