

As I tested larger and larger values of n for Hanoi. I seemed to notice that every time I increased n by 1, the runtime almost doubled. While I haven't seen this kind of growth in a function before, it does make sense that the Hanoi function grows like that since each call calls the recursive function another two times. So therefore, I would say that the Big O is 2^n .

To switch between the different kinds of deques using the command line, we'd have to add space for a string argument for the name of what kind of deque we want our program to use. That was we could just use a function like `get_deque` and instead of checking if the input is equal to a certain numerical value, we can instead check for a string value like "array" or "linked_list" from a certain indexed argument.

To test Hanoi, I ran the function and looked at what was specifically in the Stacks. I looked to see if any of the stacks weren't in the proper order in terms of where order the pegs were. Since my output for $n=3$ was also matching the project descriptions output, this meant that my output was probably right and was using the right logic to move around the pegs.

To test the delimiters, I made my own text file for tests with grouping symbols in them. I checked several of both cases that worked and ones that didn't work to see if the delimiters were being checked properly.