Can someone check my understanding of this algorithm within the overall context of creating a DPA graph?

I've read through all the posts and I've created a graph according to my understanding, but the algorithm takes a little long (~3 min) and the graph looks a bit wonky.

1. We create a completely connected graph containing m nodes (\_\_init\_\_ portion of DPA class)

2.  We add n-m nodes, which if labeled by their node number would be nodes m to n

  --  for each of these nodes we need to run a trial to randomly select which nodes already within the graph, will be connected to the new node (some node number between m and n)

  -- these randomly selected nodes are then added to the graph

  -- we repeat this every time, and as a consequence there is a possibility for a greater number of nodes to be added to each successive node (right)

so 1. is covered in the initialization of the DPA class, #2. is something we implement, but all the sub points of #2 are covered within the trial, correct?

Thus following completion of this algorithm we have n nodes, m of which are completely connected, n-m of which are randomly collected to some subset of the previous.

Our task for the problem is to determine the in-degree distribution of the graph at completion. All this sound right? If not please let me know