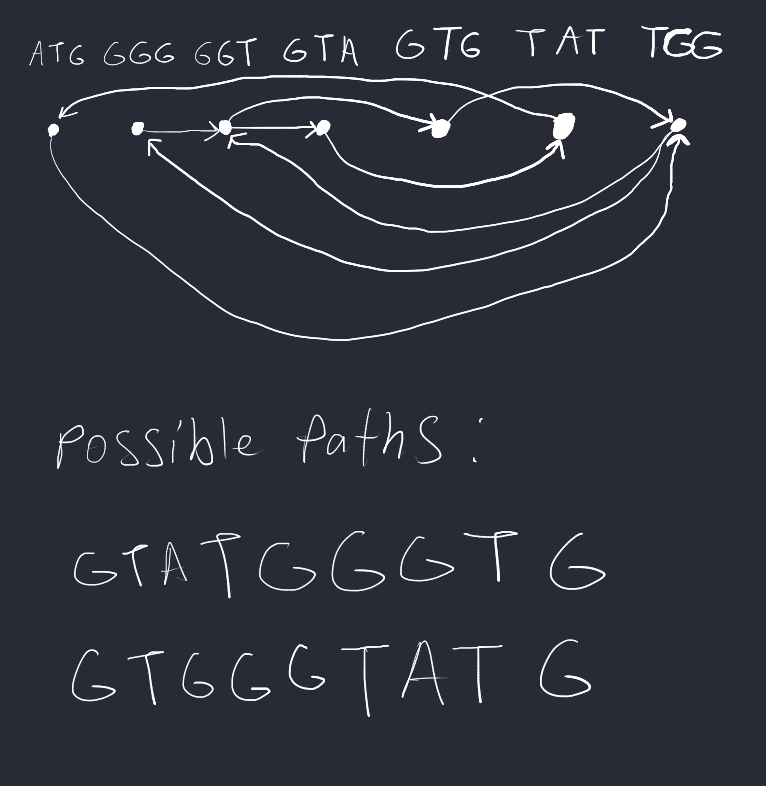
CAP 6515 HOMEWORK ASSIGNMENT 4  
DUE ON 11-22-2022  
Note: Any solution to an algorithm design question MUST contain the following  
four sections:  
(1) Problem statement. A clear unambiguous statement of the problem to  
be solved, which includes the input, the output, and the object function  
with the constraints.  
(2) Algorithm description. A clear, unambiguous description of the algo-  
rithm.  
(3) Correctness proof. A convincing mathematical argument that the algo-  
rithm described solves the computational problem described.  
(4) Time analysis. A time analysis of the algorithm, up to order, in terms of  
all relevant parameters.  
You may use any algorithms and data structures from class.

1. Hamiltonian Path and Eulerian Path (100 pts)  
Use both the Hamiltonian path approach and the Eulerian path approach  
to solve the sequence assembly problem for the following spectrum: S={ATG,  
GGG, GGT, GTA, GTG, TAT, TGG}.  
Please label the edges and vertices of both graphs, and give all possible sequences  
that can be explained by the spectrum.

Hamiltonian:

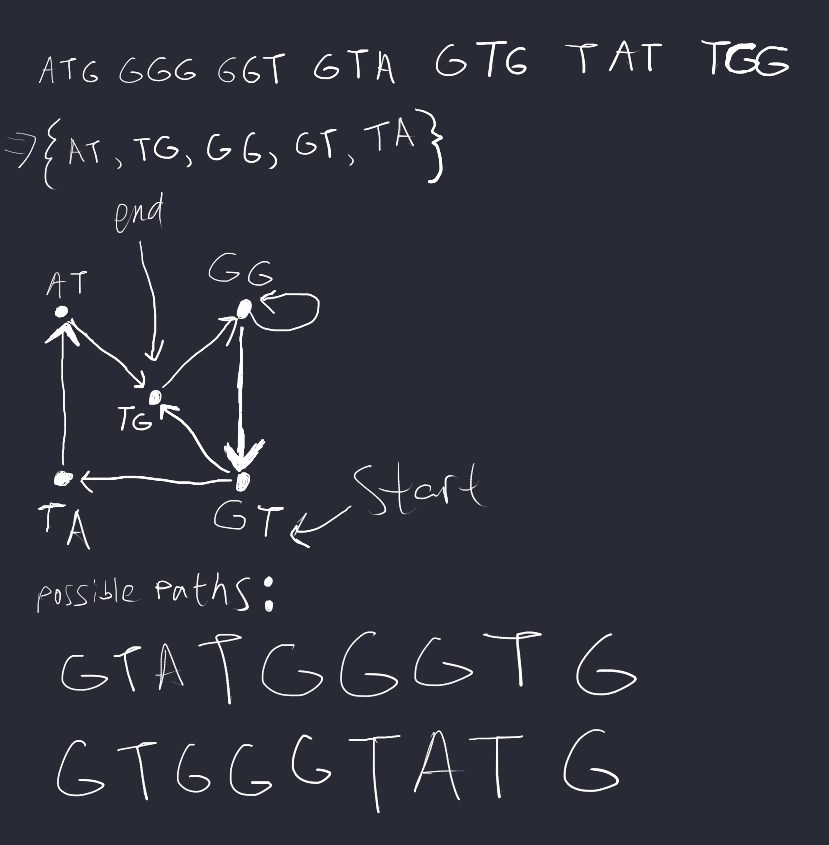
For this approach, we must visit every node in the graph exactly once. The graph is drawn in such a way that overlapping strings will have a directed edge going from one node (3-mers) to the other node representing the strings.



The time complexity would scale exponentially with our input n for producing one possible string combination because we traverse every node in the Hamiltonian approach and given n strings, we can generate many strings from every node. This is an NP-Complete problem.

Eulerian:

For this approach, we must visit every edge in the graph exactly once. The graph is drawn in such a way that overlapping strings will have a directed edge going from one node to the other node representing the strings. The given strings of three characters long are broken down into two characters. For example, ATG would yield two nodes (2-mers), AT and TG with the node AT pointing to TG. In the graph, the starting node is the node with more outgoing edges than incoming ones, that is why GT is the starting node. The ending node is the node with more incoming edges than outgoing ones, that is why TG is the ending node.



For Eulerian cycles, the time complexity would scale linearly with the input n for producing one possible string combination because we traverse every edge in the Eulerian approach and given n strings, we created n edges in the graph.