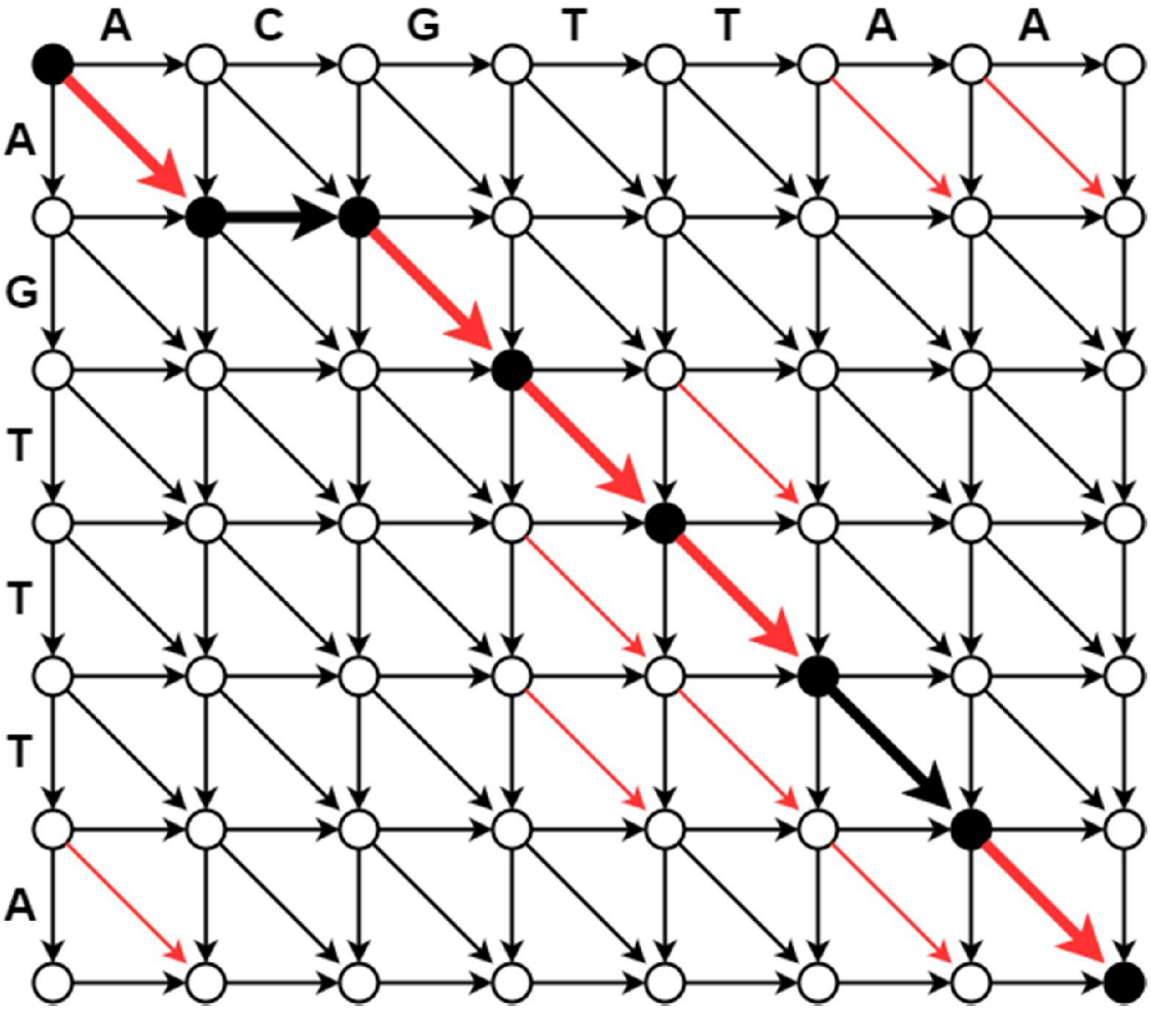
1) AGCAGCTGCTGCA
2) To visit every edge once, starting an a node with an indegree or outdegree could leave you stuck with some edges unvisited if when you come back to that node. Starting from a node with a high indegree or outdegree would solve this problem, since when you return to the node you guarantee that more of the edges would be traversed. However, starting at such a node would not guarantee that the first cycle found is an Eulerian cycle, as the edge that leads to the Eulerian cycle could be any of the many outgoing edges, while a node with only one outgoing edge would guarantee that if there is an Eulerian cycle, that edge has to be in the cycle.
3) FindBubble(start):
   For each edge from start:
     Make set of all nodes reachable from endpoint of edge
   If all the sets are pairwise disjoint
     No bubble
   Else:
     Bubble from start to each node in set of each pairwise intersection
   You should not remove either bubble, unless the application calls for it. If we had to choose, which path to remove would depend on the use case.
4) The algorithm is correct, and has a runtime of |Spectrum|^2 + MaxMass.
   SpectralConvolution(Spectrum):
   Sort(Spectrum)
   Init array A of |Spectrum| * |Spectrum|
   For each M in Spectrum:
     For each M1 > M in Spectrum:
       A[M1][M] = M1 – M
   Flatten and sort A
   Count multiplicities in A and return top
5) Yes. If you take a cyclic peptide as a linear peptide, its theoretical spectrum is the same as the corresponding linear peptide. However, because it is cyclic, its spectrum will also contain the fragments that roll over the ends, thus it will be a superset of the linear peptide's spectrum.
6) CountWithGivenMass(mass): // recursive but has same effect as bottom-up dynamic programming since each CountWithGivenMass calculated once
   A = static array of amino acid masses
   B = static array of previously found answers
   If B[mass] != null:
     Return B[mass]
   If mass < min(A):
     B[mass] = 0
     Return 0
   Else:
     If mass in A:
       Sum = 1
     Else:
       Sum = 0
     For m < mass in A:
       Sum += CountWithGivenMass(mass – m)
     B[mass] = Sum
     Return Sum

7) FindParentMass(Spectrum):
```
  A = SpectralConvolution(Spectrum)
  B = empty_set
  For E in A:
    For M in Spectrum:
      B += E + M
  Return largest value in B with highest multiplicity
```
8) Optimal alignment is:
ACGTTAA
A_GTTTA
The path and graph are shown below.



9) When constructing the graph to traverse through, add zero-cost edges from the start node to each node in the top row, and zero-cost edges from each of the nodes in the last column to the end node. Assuming that u is going across and v is going downwards. Then, run the algorithm as usual.
10) No, for example, the optimal multiple alignment of CCCCTTTT, TTTTGGGG, and GGGGCCCC does not produce any of the optimal pairwise alignments.