Dennis Kuzminer
CSCI-UA 310-001 PS3b

3. To see if there is a k-alternating path, we can
   a. Making k copies of the graph ($G^{(0)}$, $G^{(1)}$... $G^{(k-1)}$, $G^{(k)}$) connected by k edges with each edge that connects two graphs alternating between black and white.
   b. Particularly, connect two such copies, i and i+1, iff an edge from i to i+1 (using the same connections as in the original graph) changes the current path color (alternates). This will make the "new" connections to the graph alternate in color.
   c. Run Dijkstra from some node in $G^{(0)}$ to some node in $G^{(k)}$. Record whether Dijkstra has found a path.
   d. Run Dijkstra again from some node in $G^{(0)}$ to some node in $G^{(k)}$; however, the starting node this time should have a different color than the staring node in step c. Record whether Dijkstra has found a path. This is to check whether there is a k-alternating path but starting on the opposite color of $G^{(0)}$.
   e. If Dijkstra has found a path on either of these, then there exists a k-alternating path.

   In this case, because we are making k (with k times as many edges and vertices) copies of the graph, the time complexity of this algorithm is $O((k|V| + k|E|)log(k|V|))$. Although, we are running Dijkstra twice that should not impact time complexity.