1. Give the order in which Dijkstra’s Algorithm would visit each vertex starting from vertex A, where “visiting a vertex v” means “relaxing all of the edges out of v.”

   \[ A \quad B \quad C \quad D \quad F \quad H \quad G \quad E \]

2. Change one of the weights in the graph so that the shortest paths tree returned by Dijkstra’s is not correct. *Hint: We showed in class that Dijkstra’s shortest paths tree is correct so long as all edges are non-negative.*

   Set the weight of the edge connecting vertex \( E \) and vertex \( H \) to the integer weight \( \leq -3 \).

3. Suppose we use the following heuristic.

   \[
   \begin{align*}
   h(A, G) &= 2 \\
   h(B, G) &= 2 \\
   h(C, G) &= 20 \\
   h(D, G) &= 2 \\
   h(E, G) &= 6 \\
   h(F, G) &= 2 \\
   h(G, G) &= 0 \\
   h(H, G) &= 2 
   \end{align*}
   \]

   Recall that \( A^* \) search is just Dijkstra’s algorithm, except that the priority of a vertex \( v \) is given by the sum of the distance from the source to \( v \) plus \( h(v, G) \), and also that we stop the search when the target is visited.

   Give the path (not order visited) that \( A^* \) search returns from \( A \) to \( G \). You may not need all blanks.

   \[ A \quad B \quad H \quad G \]