3. The necessary and sufficient condition for an undirected graph to have Eulerian path is, all the vertices should have even degree. We can check if the degrees of all the vertices are even in linear time.

4. It is basically a bridge finding algorithm. Here you will find a nice article on this problem.

5. \[ |MST| = 2 + 3 = 5 \]
\[ |\text{Shortest Path Tree}_| = 2 + 4 = 6 \]

In the worst case, Shortest path tree can have weight = \(|MST|\)^2

6. Suppose a new edge \( e \), is inserted in between \( u \) and \( v \). This \( e \) will create a cycle in MST. If \( e \) is the highest weighted edge on that cycle, then the previous MST will still be the MST of the new graph. Otherwise, just removing the highest weighted edge from the cycle will give the new MST.
Suppose a new edge e, is inserted in between u and v. If d[u] + |e| > d[v], the previous tree will still be the shortest path tree of the new graph. Otherwise, d[v] will change to d[u]+|e|. Now, for other nodes x, if x is reachable from v, we relax d[x].