1. Design an algorithm for finding the number of distinct shortest paths from a given starting node s in a given unweighted, undirected graph $G = (V, E)$. Give proof of correctness and running time analysis.

2. You are given a weighted, directed graph $G = (V, E)$ and two nodes $s, t \in V$. The weight of an edge e is denoted by $w(e)$. The graph denotes a city road network where nodes are landmarks, edges are roads, and the weight of an edge denotes the time it takes to travel along that edge. It takes a long time to go from s to t even along the shortest time path. To fix this issue, the city authorities are thinking about adding a one road (between two arbitrary landmarks). They have figured out a set of possibilities in terms of pairs $(u_1, v_1), (u_2, v_2), ..., (u_k, v_k)$ of landmarks along with the estimated time $t(u_i, v_i)$ it will take if a road from u_i to v_i is built.

   Design an algorithm to find which of the roads $(u_1, v_1), ..., (u_k, v_k)$ to build such that the shortest time path from s to t gets minimised. Give proof of correctness and running time analysis.

3. Given a min-heap with n elements, design an algorithm to find the k^{th} minimum element of the min-heap. Give running time analysis.