1. Frequently we are interested in understanding underlying large-scale structure in networks, often taking the form of a latent-space structure. In this latent space, nodes occupy positions along some axis or axes and the presence or absence of links between nodes depends on their positions. Typically, links are more likely to exist between nodes that are closely situated in this latent space. However, we usually lack direct knowledge of how this latent space looks like, as we can only observe the network itself. To extract this latent space, we apply optimization algorithms to estimate the positions of nodes within the latent space solely based on the network's structure. Therefore, the goal of an *embedding algorithm* is to learn the nodes' positions.

   Suppose that we have a network represented by its adjacency matrix $\mathbf{A}$. Following the intuition that links are more likely between nodes that are close in the latent space, formulate an objective function for a minimization problem to learn the nodes' position x_i (for simplicity, embed the nodes in a one dimensional space). Solve the minimization problem. Interpret and discuss your solution including any (technical) problems that you encountered. Give directions on how to solve these problems. How would you extend your formulation to a multi-dimensional space?

   Hint: It may be a better choice to work with graph matrices that you can derive directly from the adjacency matrix $\mathbf{A}$. For example, we learned about the degree matrix, incidence matrix, and graph Laplacian.