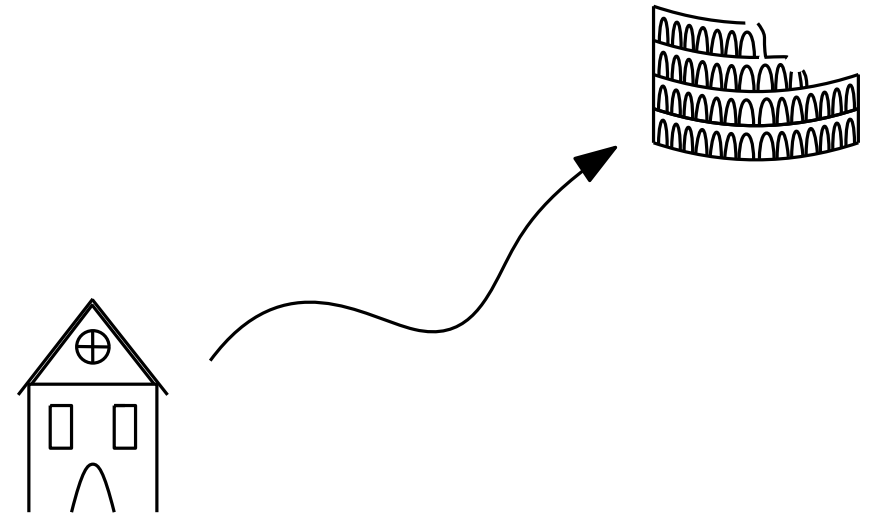


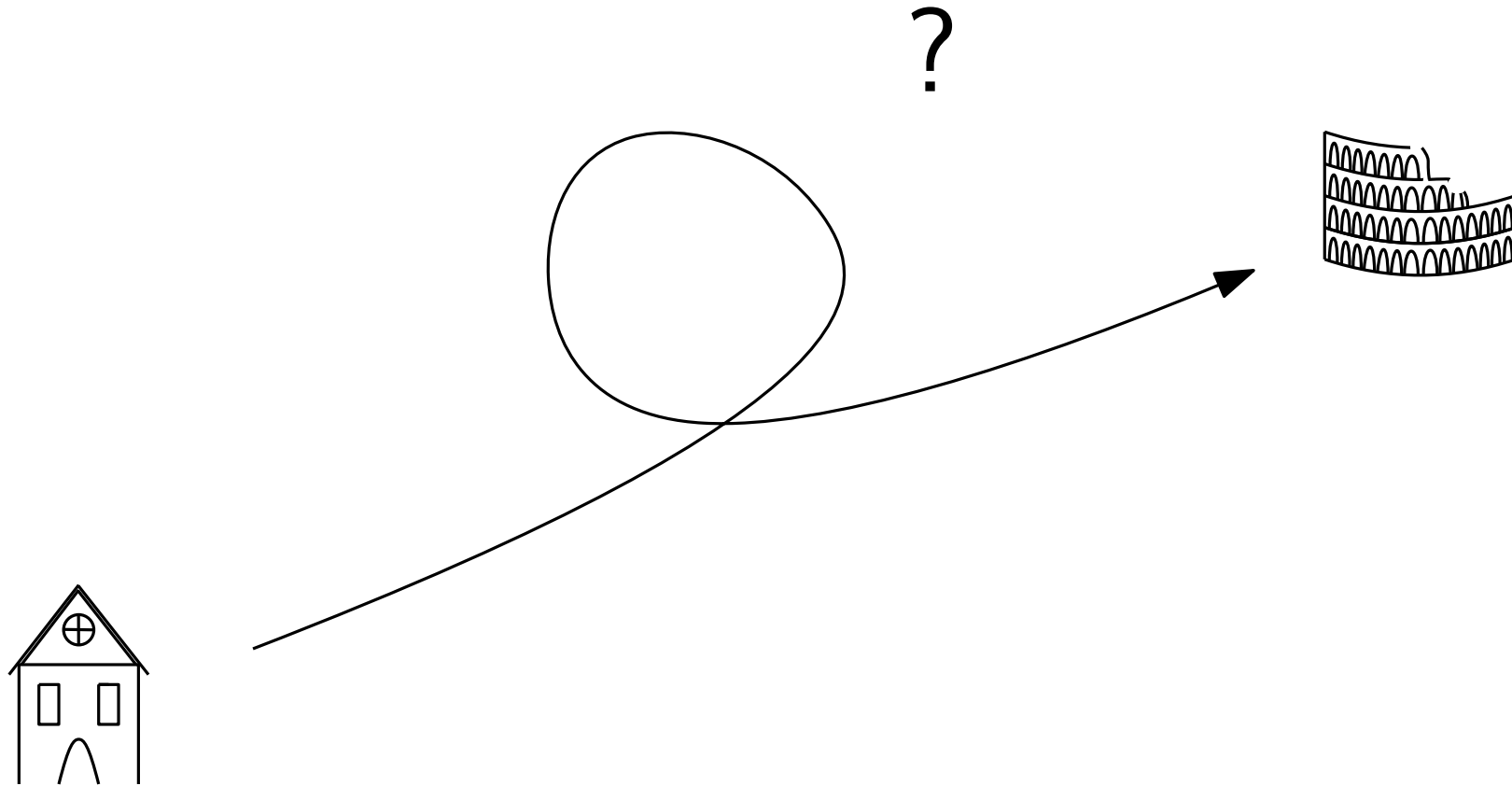
Road Reorganization

Luca Versari

WEOI 2025

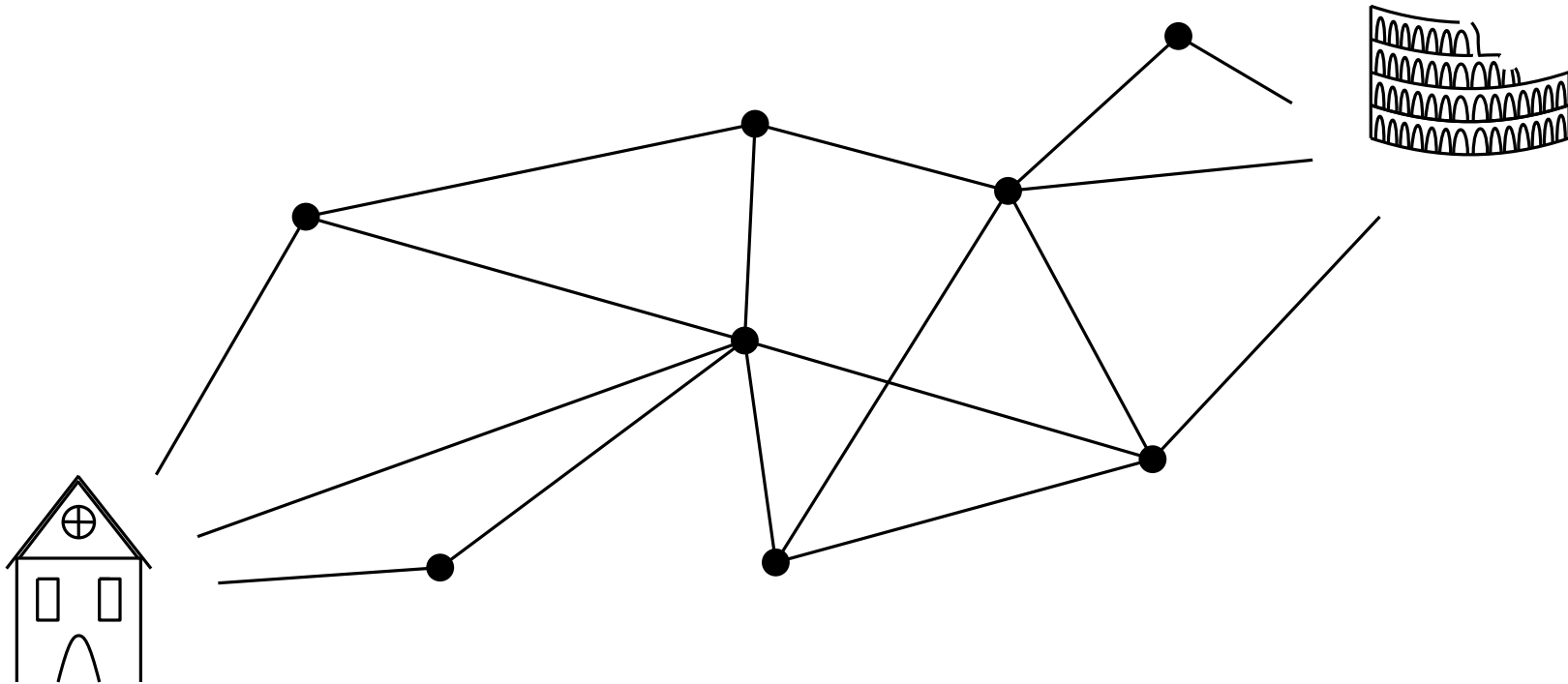


Finding the shortest path

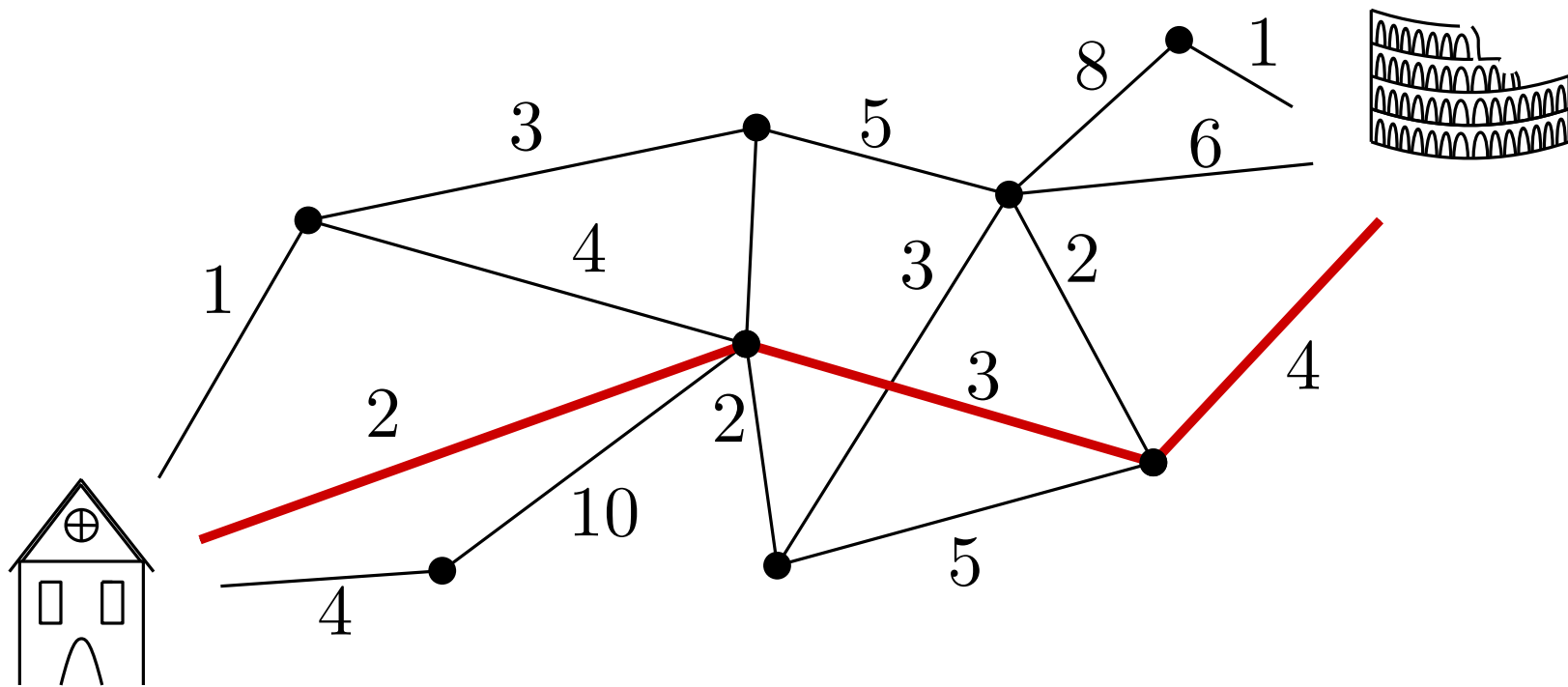


Use Dijkstra to find the shortest path in all subtasks

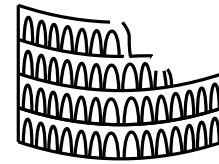
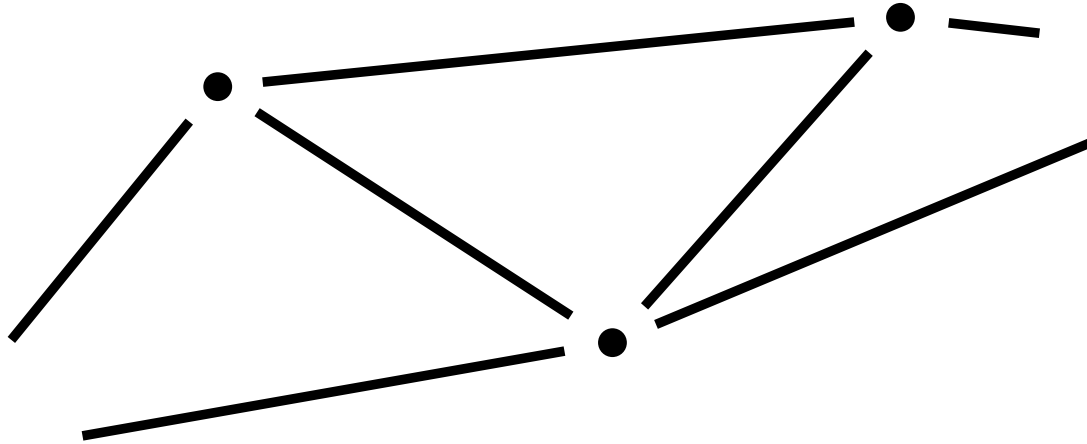
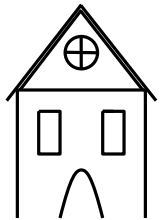
Finding the shortest path



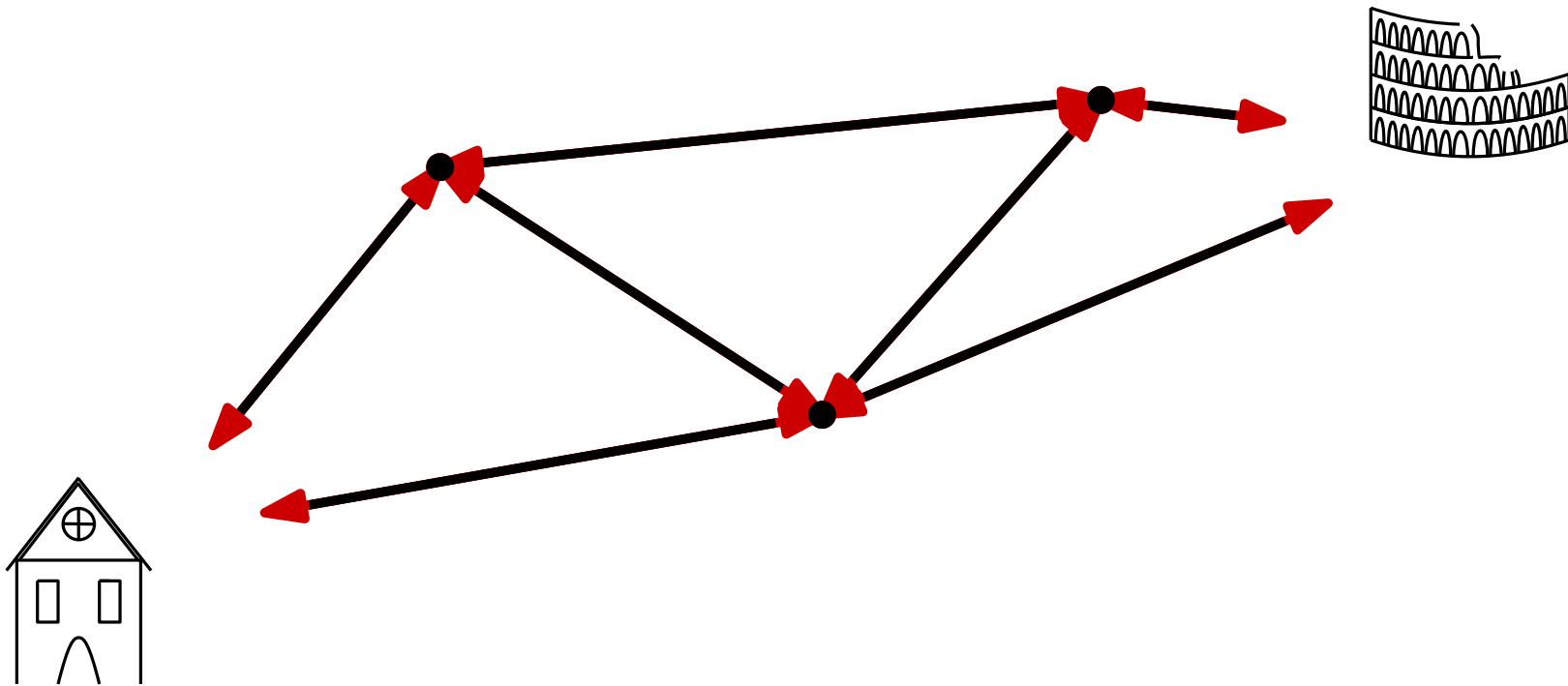
Finding the shortest path



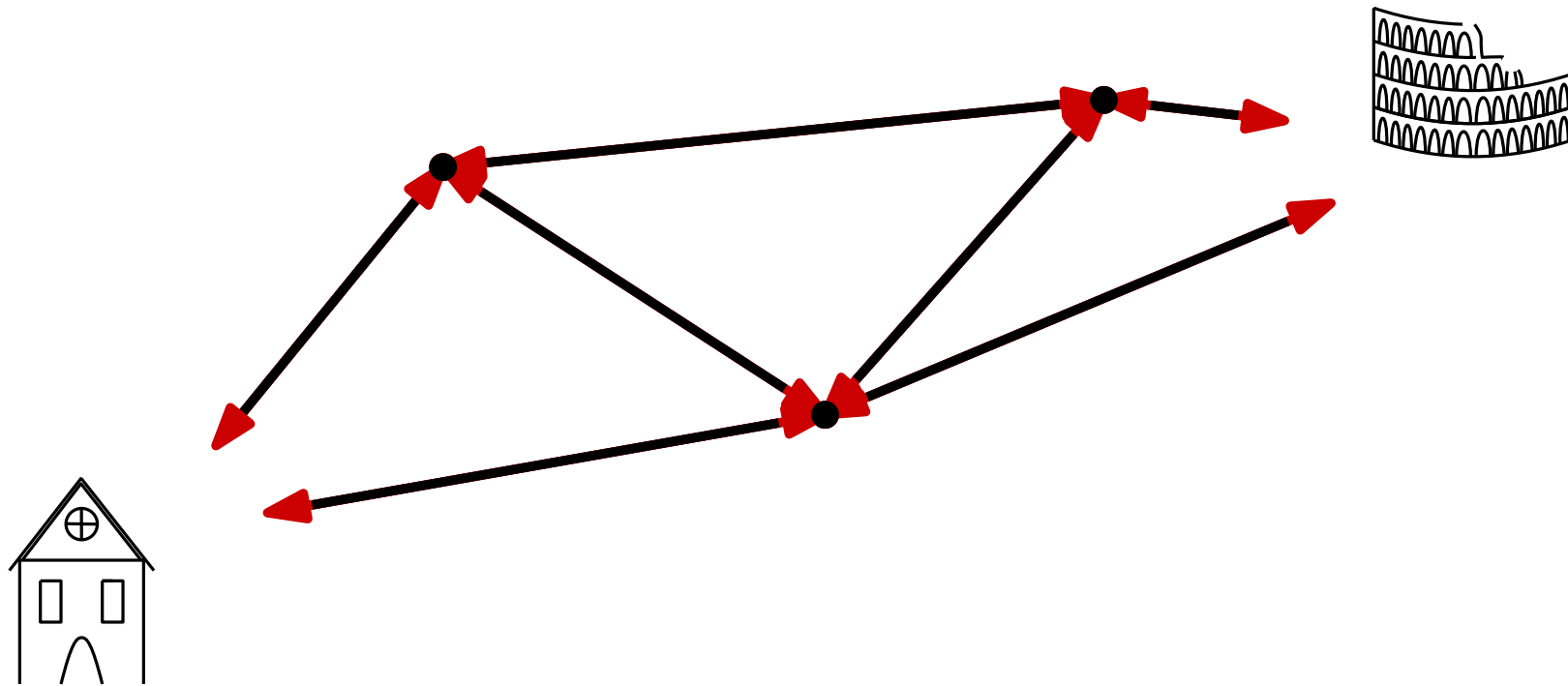
Subtask1: Brute Force



Subtask1: Brute Force



Subtask1: Brute Force



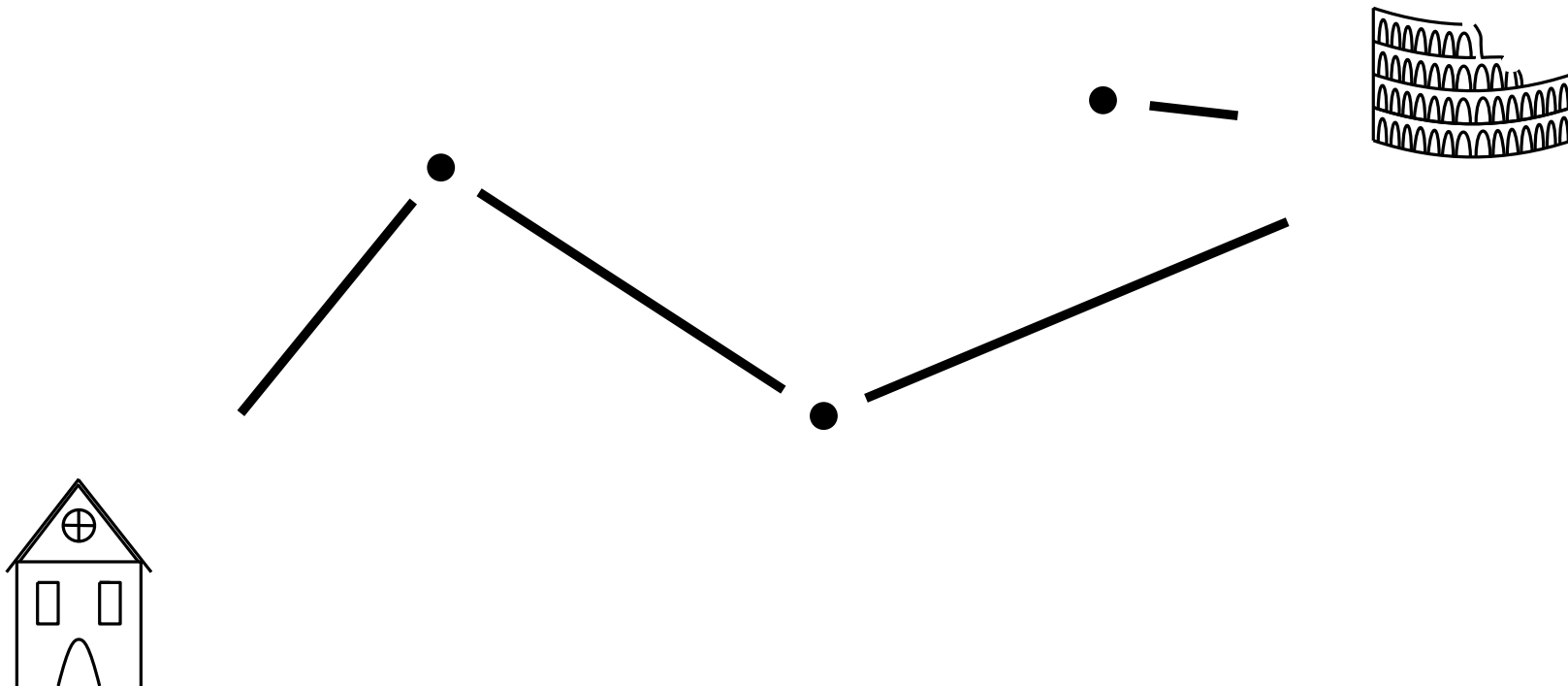
There are at most 20 edges

Try both direction for each edge gives 2^{20} options

Checking each of the options with DFS gives $\sim 20\,000\,000$ operations

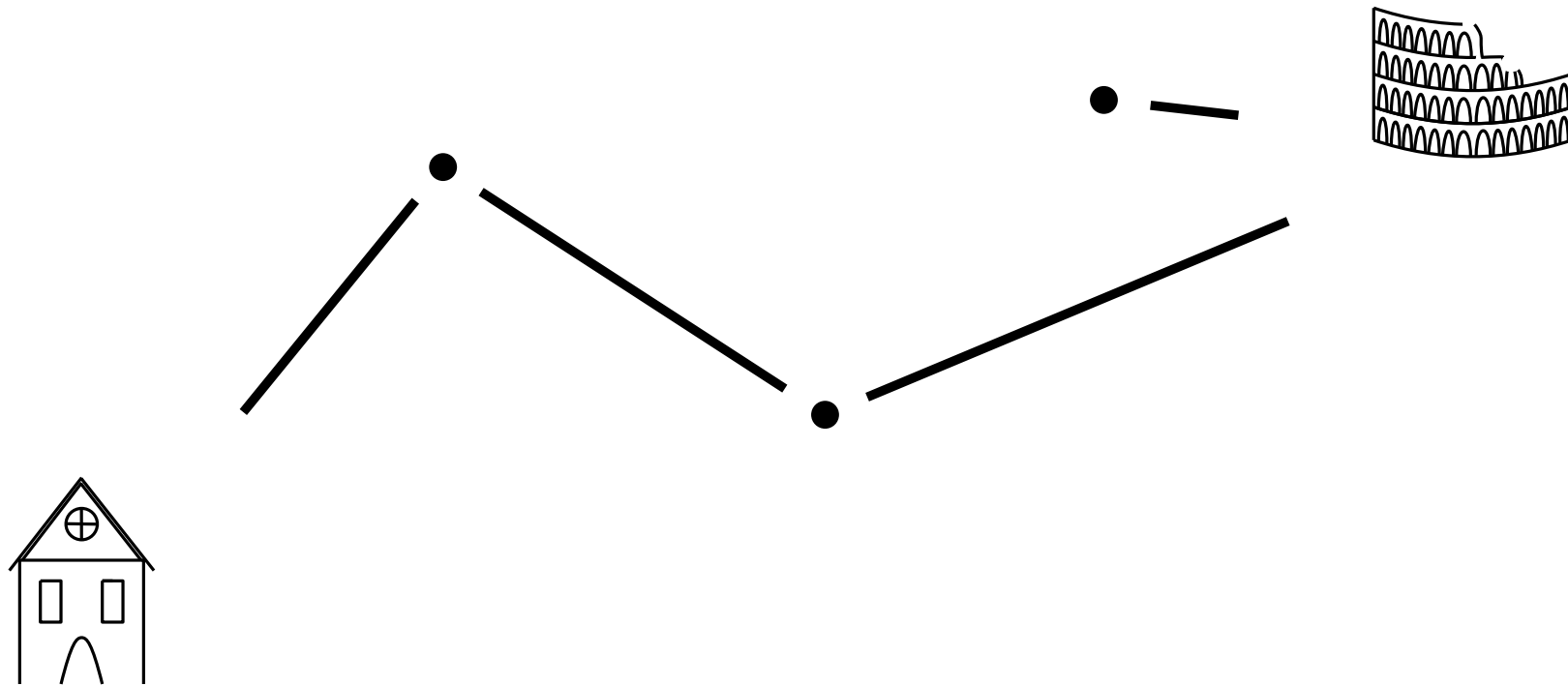
Subtask2: $N \leq M$

If the graph is a tree:



Subtask2: $N \leq M$

If the graph is a tree:



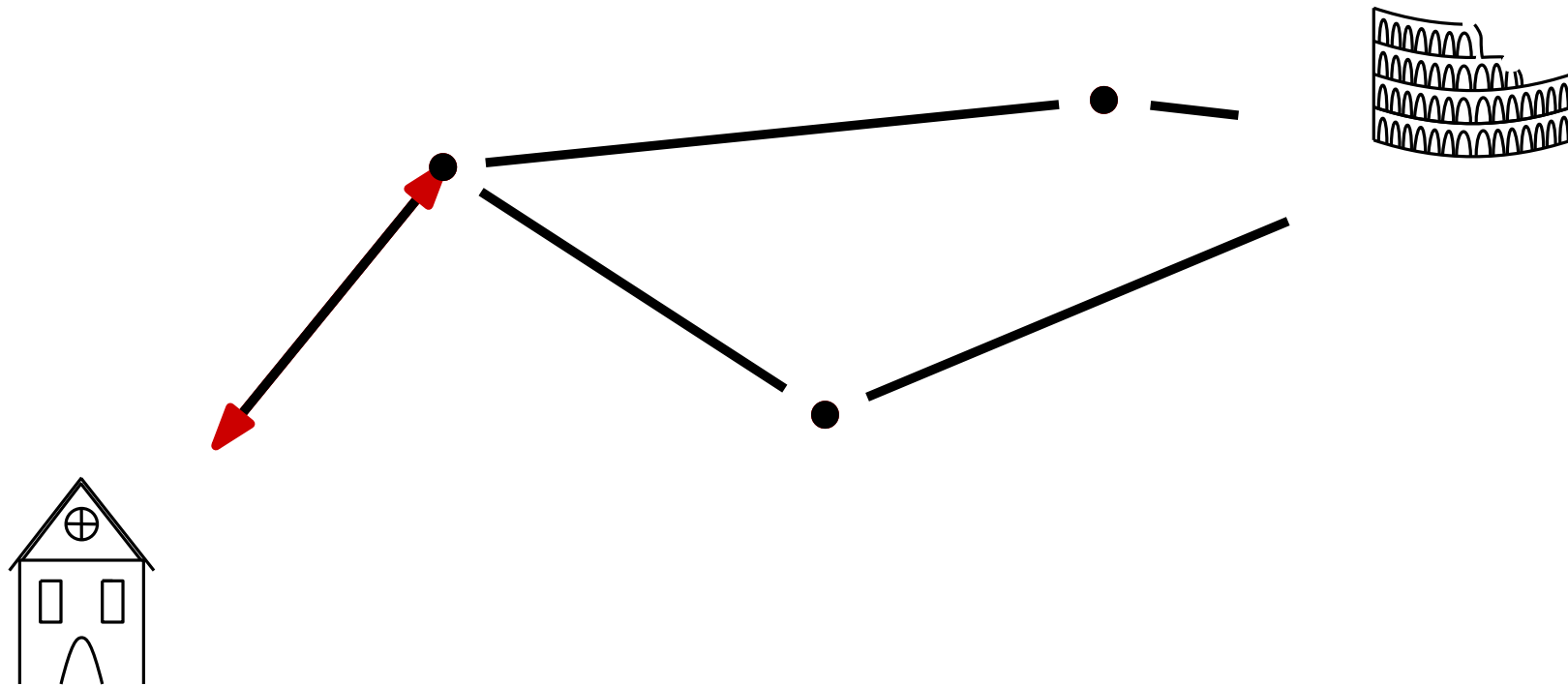
In a tree there is a unique path between each pair of vertices

Thus there is a unique path from Home to Rome, which is also the shortest

Directing this towards Rome makes it impossible to get back home

Subtask2: $N \leq M$

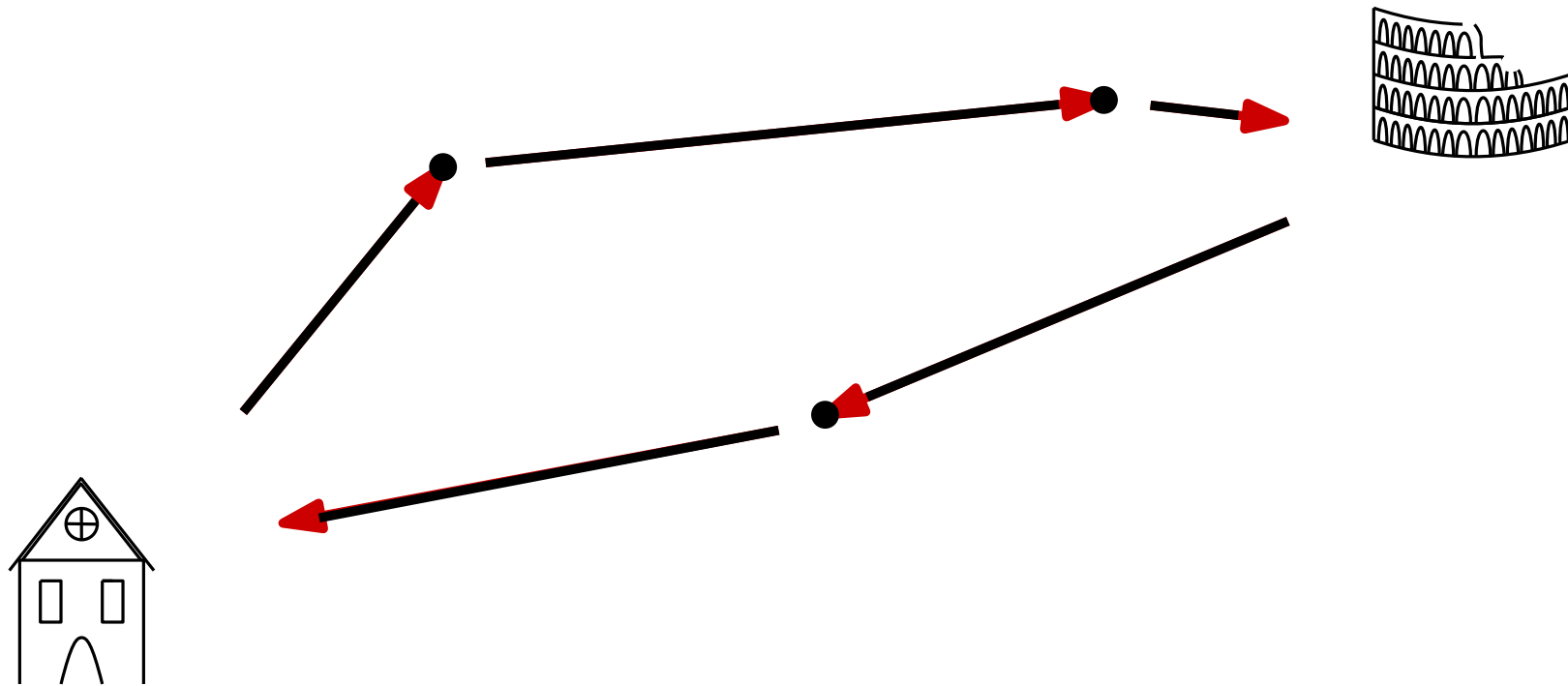
If the graph has a cycle:



If the cycle does not contain both, Home and Rome, than by the same argument as for the tree there is no solution

Subtask2: $N \leq M$

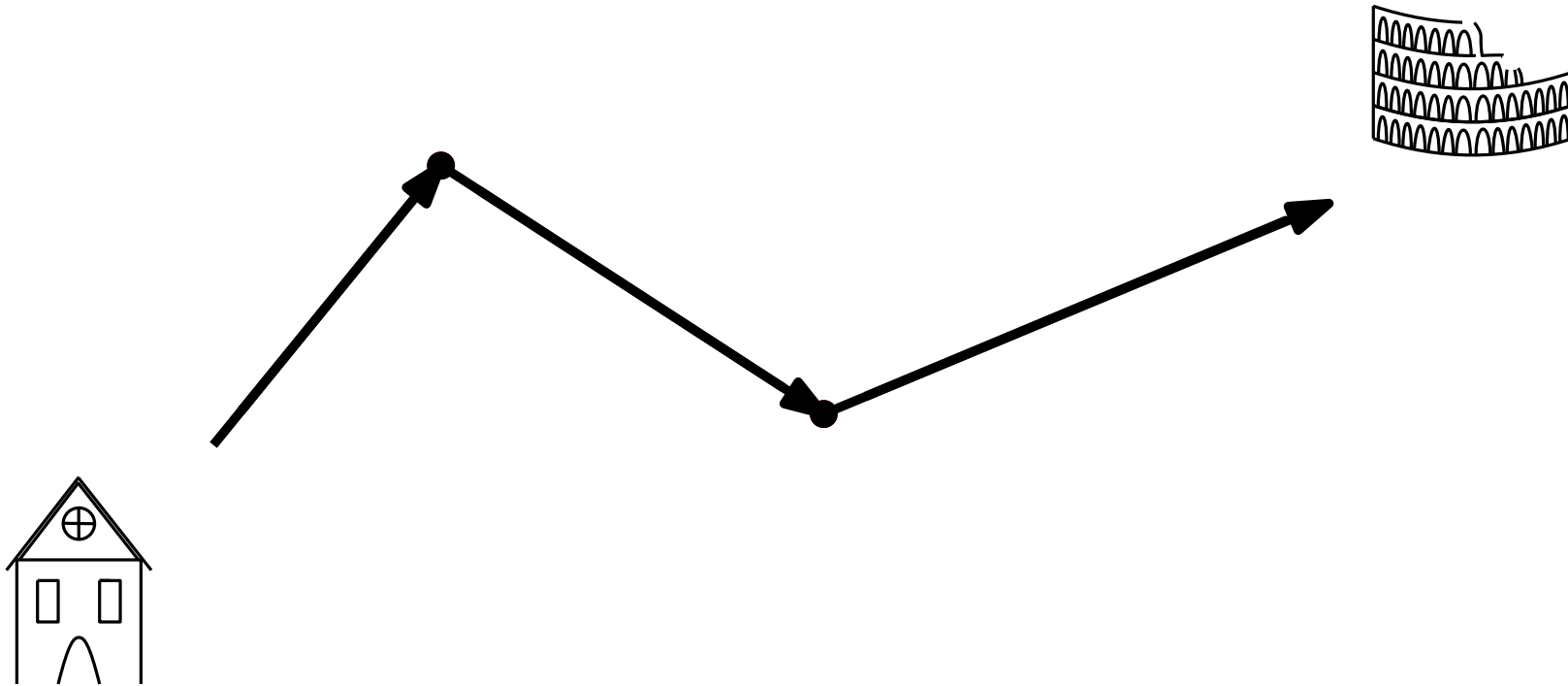
If the graph has a cycle:



If the cycle goes through both Home and Rome, then we can direct the cycle to form a directed cycle so that the shorter path goes from Home to Rome

Full solution

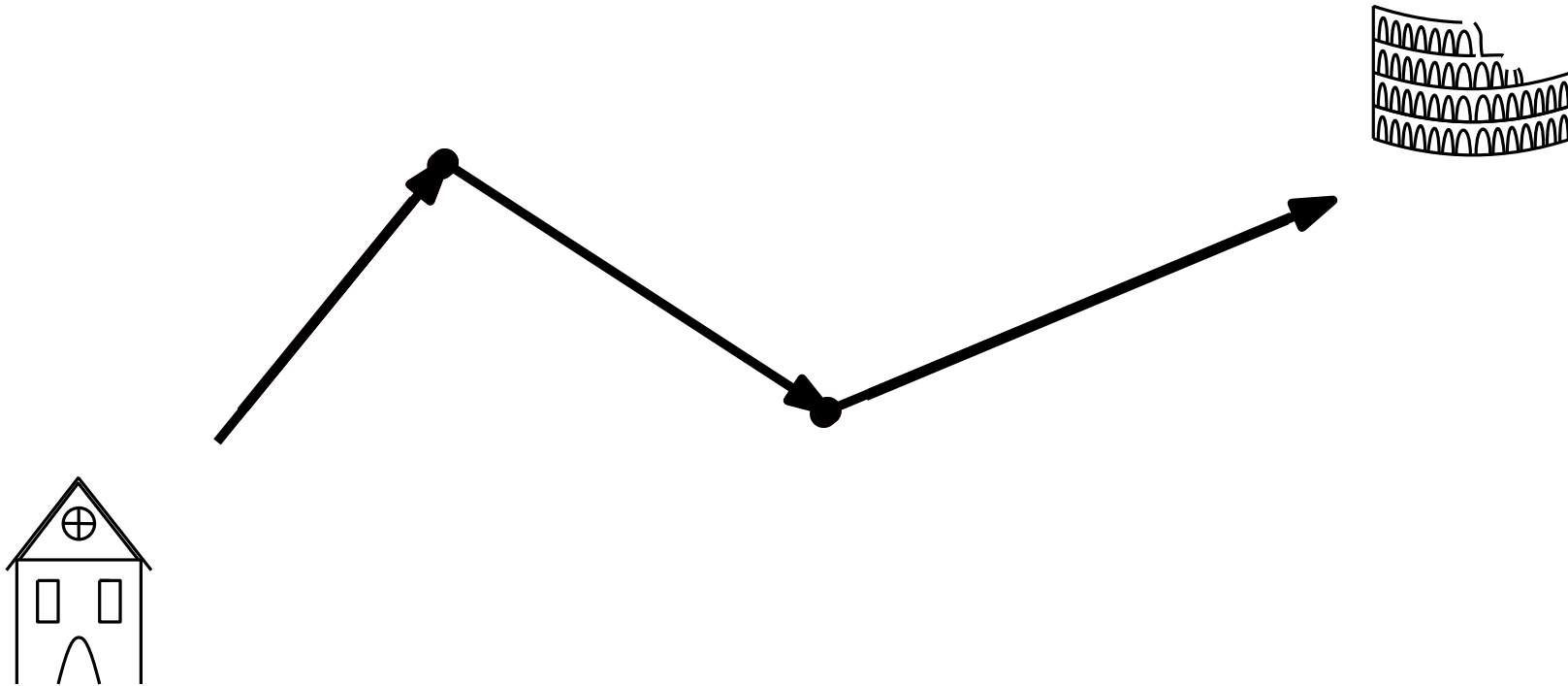
Step 1: Find a shortest path from Home to Rome



Direct these edges to go towards Rome

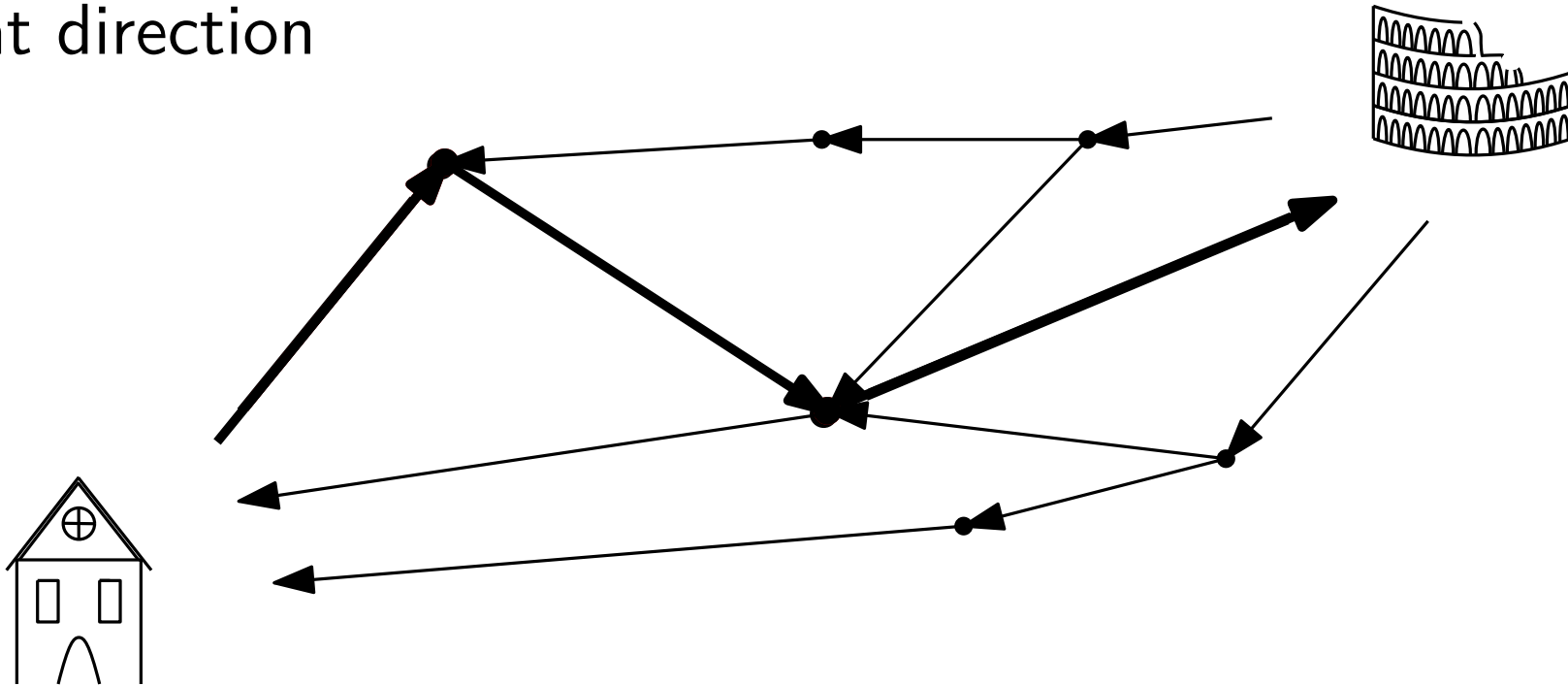
Full solution

Step 2: Direct the rest of the edges such that we can still reach Home



Full solution

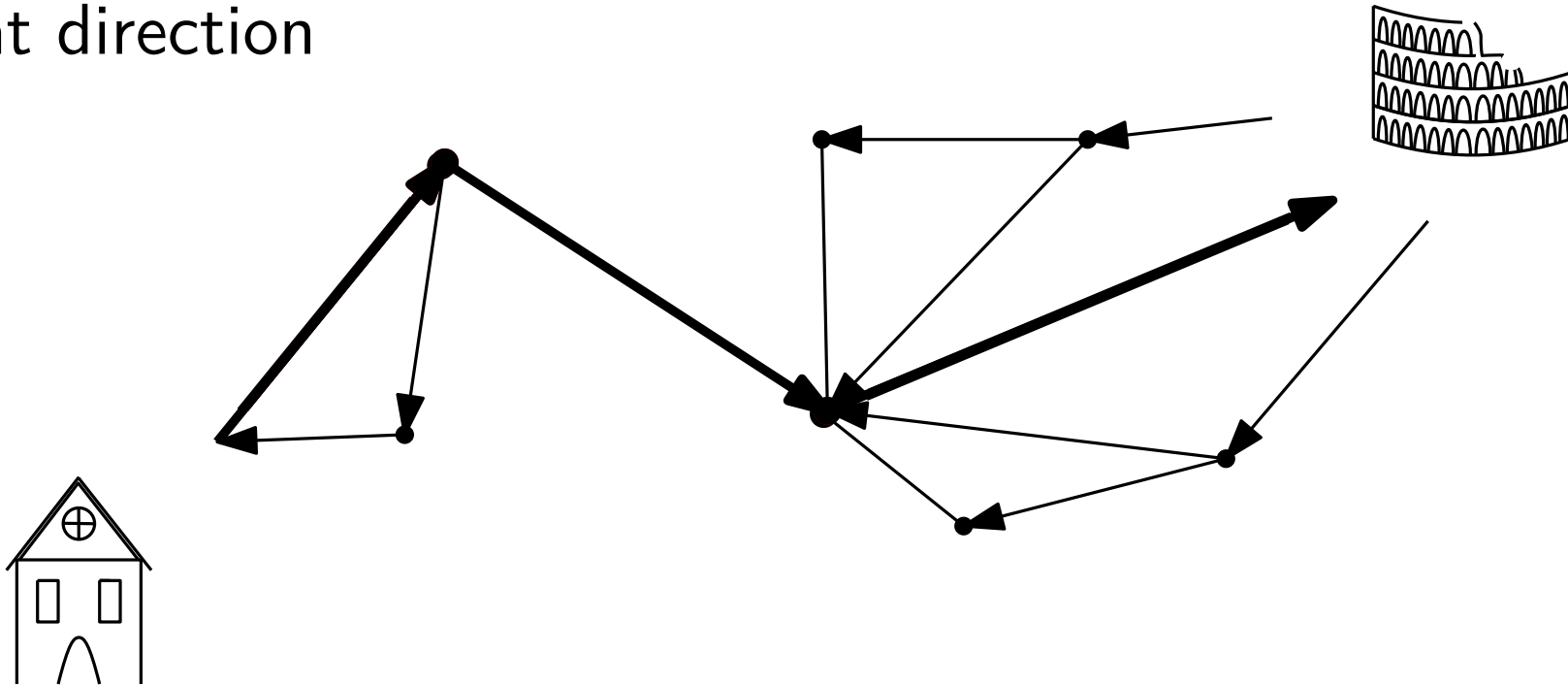
Step 2: Direct the rest of the edges such that we can still reach Home
Start a BFS in Rome, directing the edges as you encounter them, except for edges on the shortest path that are already directed and can only be used in that direction



If this reaches home, then output the edges you encountered in the direction you assigned to them and any other edges you can orient arbitrarily

Full solution

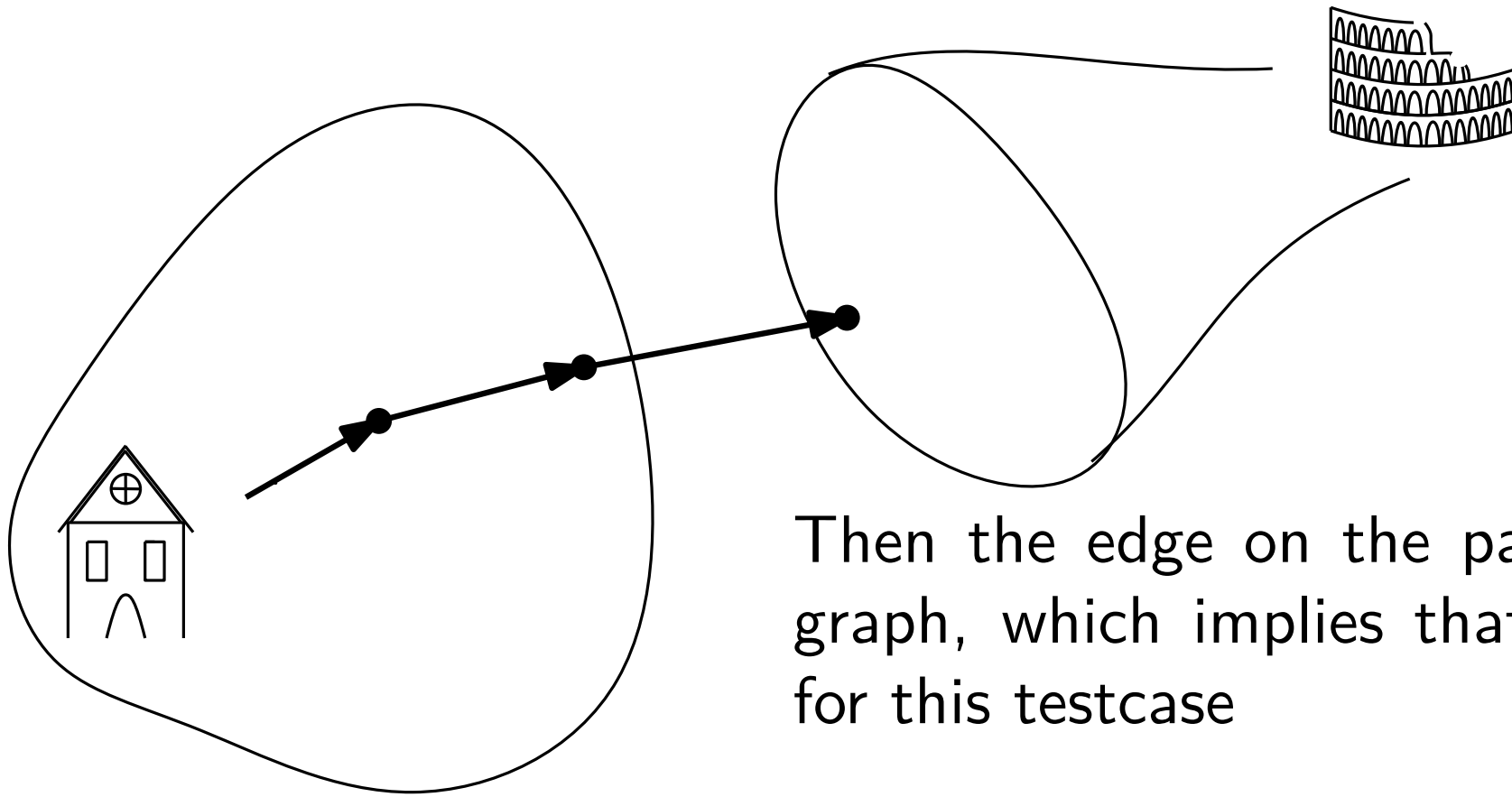
Step 2: Direct the rest of the edges such that we can still reach Home
Start a BFS in Rome, directing the edges as you encounter them, except for edges on the shortest path that are already directed and can only be used in that direction



Else there is a vertex on the shortest path such that no vertex closer to Home can be reached from Rome in this graph

Full solution

Step 2: Direct the rest of the edges such that we can still reach Home



Then the edge on the path is a bridge in the graph, which implies that there is no solution for this testcase