Problem Statement-

We are given a graph containing M nodes, along with a starting and an ending node. Amongst these nodes, N nodes are marked as special nodes. Our objective is to start from the starting node, visit each of these N nodes and then reach the end node. We have to find the shortest path for the same if possible. This problem is commonly referred to as Steiner’s TSP (Travelling Salesman Problem) and is a special form of the Travelling Salesman Problem.

Constraints-

1. Each non special node can be visited as many times as you want.
2. For the optimal solution to the problem i.e. the shortest path, each special node must be visited exactly once (Necessary but not sufficient).
3. Nodes in the graph are connected by edges and each edge can be visited as many times as you want and these edges can be traversed in either direction.
4. For the optimal solution, there must be exactly one entry and one exit into each special node (Necessary but not sufficient for Steiner’s TSP).

Incorporation into our game-

We can map the game that we have made into a case of the given problem after drawing the following parallels to the Steiner’s TSP-

(Add images over here to make it more interesting)

1. Each point in the map can be seen as a node. The total number of nodes (N) in graph is hence the size of the map.
2. Each node in the map has a value assigned to it which is either a 0 or a 1.
3. Each node has a maximum of 4 edges which represents UP, DOWN, LEFT and RIGHT nodes with respect to the current node.
4. An edge exists between 2 nodes if and only if both of the nodes have a zero value assigned to them and they are adjacent to each other in the map.
5. There are 2 flags in the game representing the host and participant base. These can be viewed as the starting and the ending node.
6. There are 6 powerups present in the game. These can be viewed as the M special nodes present in the graph. Collecting these powerups in the map equals visiting the given node in the graph that we have made.
7. The player can only move on the blocks with a value 0 assigned to them which is ensured when edges between different nodes are constrained using (4).
8. All powerups in the game have equal priority for this problem

Algorithm 1 (Greedy approach with entire maze visible) –

Assumptions-

1. The entire graph is visible to us before the simulation along with the coordinates of the special nodes as well as the starting and ending nodes.
2. We can traverse through the graph as many times as we want to find out the shortest path.

Algorithm-

1. Since, we know the coordinates of all the special nodes as well as the entire graph, we can calculate the shortest path amongst all possible pairs of special nodes along with the ending and starting nodes.
2. (Djikstras ka likho)
3. After having calculated these values and storing them, we then use a greedy approach to solve the problem.
4. We start from the starting point and find the point which is closest to the starting point.
5. We then trace the path to that point and then find the point which is closest to the current point that we are on and move to that point.
6. In this way, we find a greedy approach to reach the end point. The solution is not guaranteed to be optimal but it would be a good solution that solves the problem.

Time Complexity-

1. Let the number of nodes be N and the number of special nodes be N.
2. Djikstras algorithm has a time complexity of O(E\*log(V)) where E is the total number of edges in the graph and V is the total number of vertices.
3. Since, each vertice can have at most 4 edges, the edges cannot be greater than 4V. Secondly, we need to apply this algorithm for each special Node to get its shortest distance from all the other special nodes. So the overall time that we need to spend in this operation is O(M\*Nlog(N)) .
4. Having calculated this, we use the greedy approach to determine the path that we need to choose which takes O(M^2) time. So, the overall complexity is O(M^2 + M\*Nlog(N)) which is O(M\*Nlog(N)) since M<N.

Algorithm 2 (First Person Based)

Assumptions-

1. The entire maze is not visible to us at the beginning.
2. The simulator can only see the nodes adjacent to it and it discovers the maze as it goes.
3. The coordinates of the starting, ending and special nodes are known to us initially.