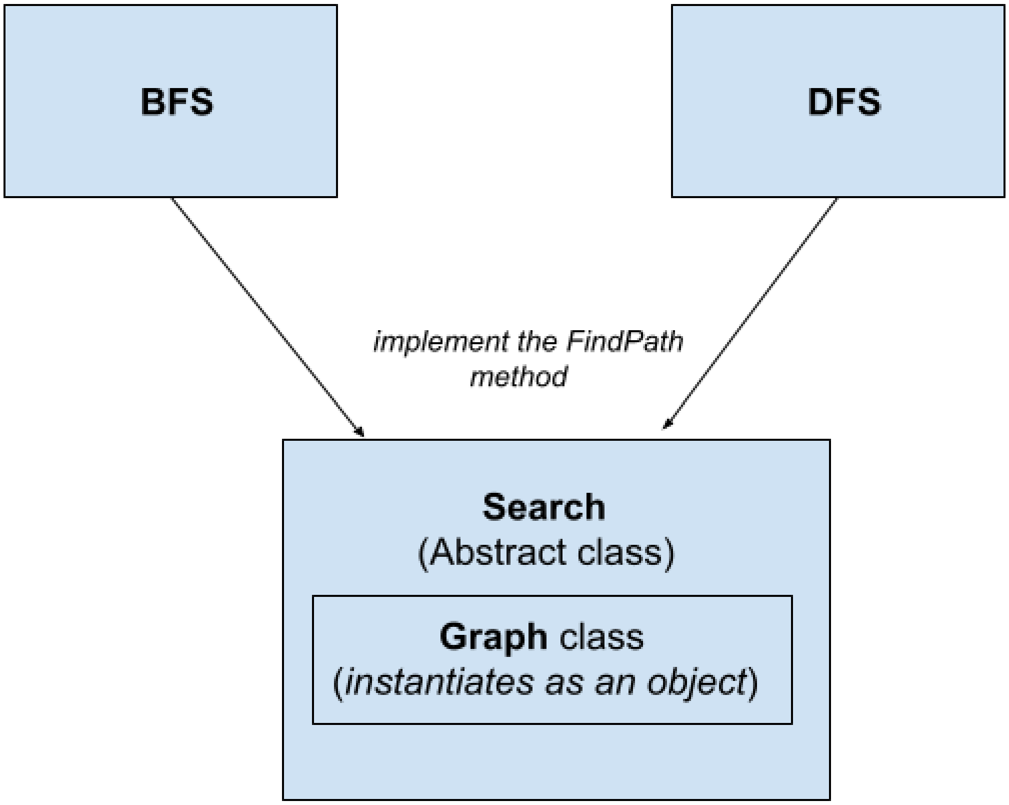
Simplified EleNa: Design Document

The aim is to compute the shortest path between the two vertices in a directed graph. The graph is built randomly and the maximum outdegree of any vertex is restricted to 10. The graph contains 10000 nodes in case of BFS algorithm and 10 nodes in case of DFS algorithm.

The time complexity to compute the shortest path using **BFS is O(V+E)**, where V and E denote the number of nodes and number of edges respectively in the randomly generated graph. **The major assumption taken to bring down the complexity is that the edge weights of all the edges in the graph are taken to be the same, for instance k.** In this way, our BFS algorithm becomes the same as **Dijkstra's** algorithm for finding the shortest path.

Using **DFS**, the time complexity is **$O(V^E)$** due to **backtracking** where each visited node is marked unvisited to explore all the possible paths that exist between the source and the target, and then computing the minimum path of those. The time complexity is quite large and hence we constraint the number of vertices to be 10 in this case.

Architecture



BFS

DFS

implement the FindPath method

Search
(Abstract class)

Graph class
(*instantiates as an object*)