**Breadth-first search (BFS):**

* **Idea:** BFS is a graph traversal algorithm that starts from the root of the graph and expands all of the nodes and all of their successors layer by layer until a goal is reached. Early stopping means that if a successor of a node is a goal, the search concludes.
* **Algorithm:** The algorithm implements two arrays: One is a list of nodes waiting to be explored called a ‘frontier’, the other is a dictionary of node state and node to prevent the algorithm from looping itself indefinitely, called ‘reached’. The search starts by checking whether the initial state is a goal or not, after which that state is added to the queues. Then a while loop iterating through each node of the graph (in this case, each ‘polygon’ of the ‘city map’) is initiated. For each iteration, the algorithm retrieves the first element of the frontier and expands that element. Each successor of that polygon is applied a goal test for early stopping. Then, if that child is the goal, the program returns; otherwise, its state is checked whether a similar state exists inside the ‘reached’ dictionary to append to the end of the frontier, waiting to be retrieved in a future iteration.
* **Pseudo-code:** Taken from lecturer’s slide:

A screenshot of a computer

Description automatically generated

* **Example run:**