Algorithm 1 Breadth First Search (Graph)

```
1: function BFS(problem) returns a solution, or failure
2:     node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
3:     if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
4:
5:     frontier ← a FIFO queue with node as the only element
6:     explored ← an empty set
7:
8:     loop do
9:         if frontier = ∅ then return failure
10:
11:        node ← POP(frontier)
12:        add node.STATE to explored
13:
14:        for each action in problem.ACTIONS(node.STATE) do
15:            child ← CHILD-NODE(problem, node, action)
16:
17:            if child.STATE ∉ explored or frontier then
18:                if problem.GOAL-TEST(child.STATE) then return SOLUTION(child)
19:                fontier ← frontier.APPEND(child)
```