**function** BREADTH-FIRST-SEARCH(*problem*)

**returns** a solution *node* or *failure*

*node* ← NODE (*problem*.INITIAL)

**if** *problem*.IS-GOAL (*node*.STATE) **then return** node

*frontier* ← a FIFO queue, with *node* as an element

*reached* ← {problem.INITIAL}

**while not** IS-EMPTY (*frontier*) **do**

*node* ← POP (*frontier*)

**for each** *child* **in** EXPAND (*problem, node)* **do**

*s* ← *child*.STATE

**if** *problem*.IS-GOAL(s) **then return** *child*

**if** *s* is not in reached **then**

add s to *reached*

add child to *frontier*

**return** *failure.*