

Depth First Search (DFS) – Non-Recursive

DFS(s,problem)

if problem.goaltest(s.state) /* if s is a goal */

then return problem.solution(s)

else

frontier = (s)

loop

if empty(frontier) then return FAIL

else

node = first(frontier)

remove node from frontier

for each a in problem.actions(node) do

s' = child(a,node) /* apply(a,node) */

if problem.goaltest(s'.state)

then return problem.solution(s')

else frontier = add s' to front of frontier

Breadth First Search (BFS)

BFS(s,problem)

if problem.goaltest(s.state) /* if s is a goal */

then return problem.solution(s)

else

frontier = (s)

loop

if empty(frontier) then return FAIL

else

node = first(frontier)

remove node from frontier

for each a in problem.actions(node) do

s' = child(a,node) /* apply(a,node) */

if problem.goaltest(s'.state)

then return problem.solution(s')

else frontier = add s' to **end** of frontier

Depth First Search (DFS) – Checking for repeated states

```
DFS(s,problem)
  if problem.goaltest(s.state) /* if s is a goal */
  then return problem.solution(s)
  else
    frontier = (s); explored = {}
    loop
      if empty(frontier) then return FAIL
      else
        node = first(frontier)
        remove node from frontier
        add node to explored
        for each a in problem.actions(node) do
          s' = child(a,node) /* apply(a,node) */
          if problem.goaltest(s'.state)
          then return problem.solution(s')
          else
            if s' is not in explored and not in frontier
            then frontier = add s' to front of frontier
```