procedure ComputePath(){

while goal.g > min\_s’∈OPEN(s’.f){

OPEN.remove(state w/ smallest f-value in OPEN);

CLOSED.add(s);

for all neighbors n{

if (n.search < counter){

n.g = ∞;

n.search = counter;

}

if (n.g > s.g + 1){

n.g = s.g + 1;

n.prev = s;

if n is in OPEN then remove it from OPEN;

insert n into OPEN with f-value g(n) + h(n);

}

}

}

}

procedure Main(){

counter = 0;

for each cell c

c.search = 0;

while (start != goal){

counter++;

start.g = 0;

start.search = counter;

goal.g = ∞;

goal.search = counter;

OPEN = CLOSED = ∅;

insert start into OPEN with f-value start.g + start.h;

ComputePath();

if (OPEN = ∅){

print “I cannot reach the target.”;

return;

}

follow the tree-pointers from goal to start and then move the agent along the resulting path from start to goal until it reaches goal or one or more action costs on the path increase;

set start to the current state of the agent (if it moved);

update the increased action costs (if any);

}

print “I reached the target.”;

}