The A* Algorithm

Open : priority queue of search node

Closed : list of search node

```
AStarSearch(location StartLoc, location GoalLoc, agenttype Agent){
    clear Open &closed
    // initialize a start node
    StartNode.Loc = StartLoc;
    StartNode.CostFromStart = 0;
    StartNode.CostToGoal = PathCostEstimate(StartLoc, GoalLoc, Agent);
    StartNode.TotalCost = StartNode.CostToGoal;
    StartNode.Parent = NULL;
    push StartNode on Open;

    // process the list until success or failure
    while Open is not empty {
        pop Node from Open    // node has the lowest TotalCost
        // if at a goal, we're done
        if (Node is a goal node) {
            construct a path backward from Node to StartLoc
            return SUCCESS;
        }else{
            for each successor NewNode of Node {
                NewCost = Node.CostFromStart + TraverseCost(Node, NewNode, Agent);
                // ignore this node if exists and no improvement
                if (NewNode is in Open or Closed) and(NewNode.CostFromStart <= NewCost) {
                    continue;
                }else{ // store the new or improved information
                    NewNode.Parent = Node;
                    NewNode.CostFromStart = NewCost;
                    NewNode.CostToGoal = PathCostEstimate(NewNode.Loc, GoalLoc, Agent);
                    NewNode.TotalCost = NewNode.CostFromStart + NewNode.CostToGoal;
                    if (NewNode is in Closed) {
                        remove NewNode from Closed
                    }
                    if (NewNode is in Open) {
                        adjust NewNode's position in Open
                    }else{
                        Push NewNode onto Open
                    }
                }
            }
        }
        push Node onto Closed
    }
}
```