Uniform Cost Search Pseudo-Code

Uniform-Cost-Search(problem) returns a solution, or failure  
1 node <- a node with STATE = problem.Initial-State, Path-Cost = 0  
2 frontier <- a priority queue ordered by Path-Cost, with node as the only element  
3 explored <- an empty set  
4 loop do  
5 if Empty(frontier) return failure  
6 node <- POP(Frontier) #Chooses the lowest-cost node in frontier  
7 if problem.Goal-Test(node.State) then return Solution(node)  
8 add node.State to explored  
9 for each action in problem.Actions(node.State) do  
10 child <- Child-Node(problem, node, action)  
11 if child.State is not in explored or frontier then  
12 frontier <- Insert(child, frontier)  
13 else if child.State is in frontier with higher Path-Cost then  
14 replace that frontier node with child