

D. Considering the cost associated with every move to be the same (uniform cost), write a function which can backtrack and produce the path taken to reach the goal state from the source/initial state.

Reversal of path found can be done using backtracking as shown in the below pseudo code:

```
ReversePath(nodePair,explored)
Path <- List(Head(nodePair))

Parent <- Second(nodePair)
While parent is not Nil
Do path<- Cons(parent,path)
nodePair <- FindLink(parent,explored)
    Parent <- Second(nodePair)
Return path

FindLink(child,explored)
If child = Head(Head(explored))
Return Head(explored)
Else return FindLink(child,Tail(explored))
```