

backtracking 19

Sunday, 22 October 2023

9:59 AM

① The Knight's tour problem

Backtracking works incrementally and is an optimization over the Naive solution where all possible configurations are generated and tested.

Naive

Generate all tours one by one and check if generated tour satisfied the constraints.

while there are untried tours

{ generate the next tour
if this tour covers all squares
{ print this path
}

}

Backtracking

if all squares are visited
print the solution

else

- Add one of next moves to solution vector and recursively check if this move leads to a solution. (knight can make max of 8 moves)
- if the move chosen in above step doesn't lead to a solution then remove this move from solution vector and try alternative moves
- if none of alternative moves work then return false
(returning false will remove the previously added item in recursion and if false is returned by initial call of recursion then no solution exists)

② Rat in a Maze