Lab 1 report: Knight tour problem

Course: Introduction to artificial intelligence

Theory lecturer: Nguyễn Ngọc Thảo

Practice lecturer: Nguyễn Hải Đăng

Hồ Thị Thanh Tuyền

Lê Ngọc Thanh

Student name: Lê Nguyễn Nguyên Anh - 20127438

Class: 20CLC09

I/Project Requirement:

We write program to solve the knight tour problem by applying Backtracking and Warnsdorff's Heuristic, with sizes include 8, 15, 25 and with each of the sizes we have to get 5 different initial points to get data for experiment.

+For Backtracking:

Size 8: operate all 5 initial points

Sizes 15 and 25: we only run the algorithm for 1-hour-long-run because of the slowness of backtracking

+For Warnsdorff's Heuristic: run 5 inital points for all board size

II/Files:

Subfolders include:

SOURCE:

+20127542.cpp: The program run both backtracking and Warnsdorff's Heuristic algorithm at once when we call through cmd with argument (compile with g++ and run the exe file).

OUTPUT:

+Images of result in txt for a sample case.

+Txt files with format 20127438_backtrack and 20127438_heuristic include values: x, y, m, number of moves (ignore the first value because it is initial value and not cound move backtrack in the backtracking algorithm), time measured (in milliseconds) and board with knight moves.

DOCUMENT:

+A pdf file brief about the problem, statistics base on data from running operation of backtracking and heuristics (include both data board and figure),the completeness of the project.

III/Criteria:

#	Criteria	Status
1	Manipulate the input and output	Done
2	Implement the backtracking	Done
3	Implement the Warnsdorff's Heuristic	Done
4	Provide valid results for backtracking strategy	Done
5	Provide valid results for Warnsdorff's Heuristic	Done
6	Provide all the evidential	Done
7	Report sufficient information in the document	Done

IV/ Statistics and comments:

Backtracking:

+Database:

+Chart:

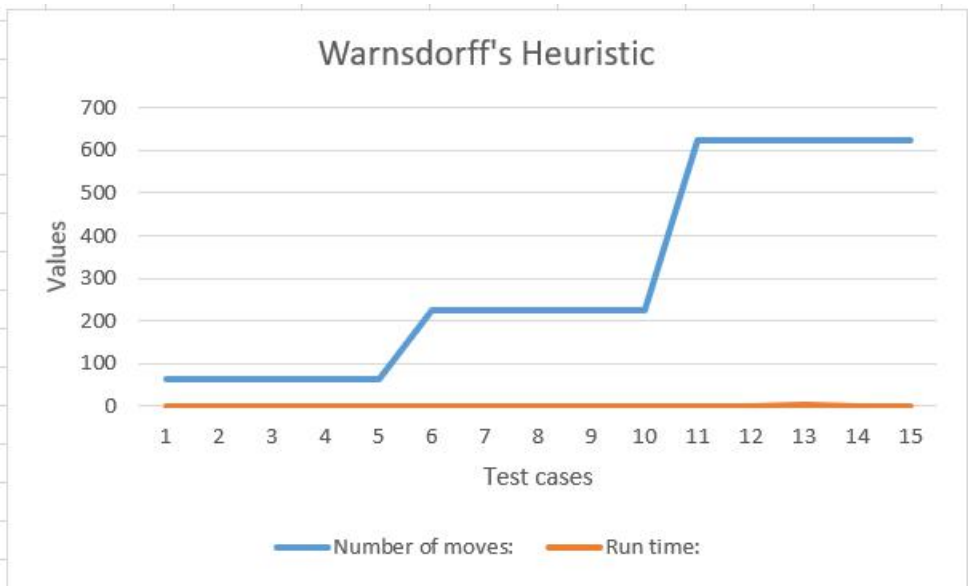
+Comment: an algorithm try exhaustedly for entire possible knight moves since it take giant space and almost infinitely long time. This method is a simpliest but it also the slowest.

Warnsdorff's Heuristics:

+Database:

Heuristic:															
(x, y, m)	(3, 6, 8)	(6, 3, 8)	(2, 7, 8)	(5, 6, 8)	(8, 2, 8)	(2, 6, 15)	(5, 3, 15)	(1, 7, 15)	(4, 4, 15)	(3, 5, 15)	(6, 8, 25)	(5 ,1 , 25)	(3, 1, 25)	(6, 6, 25)	(5, 5, 25)
Number of moves:	63	63	63	63	63	224	224	224	224	224	624	624	624	624	624
Run time:	0	0	0	0	0	0	0	0	0	0	0	1	2	1	1

+Chart:



+Comment: Thanks to heuristic moves, this algorithm solve the problem faster than backtracking because the knight will move rationally (move to pieces that have minimum unvisited adjacent). After the CSP section in class, I notice that this method like Degree Heuristic in the CSP when apply to search problem.Warnsdorff's Heuristic just need a few milliseconds to find out the solution. There are the wonderful results compared to Backtracking

V/References:

https://bradfieldcs.com/algos/graphs/knights-tour/

https://github.com/wisn/knights-tour/tree/master/outputs

https://www.geeksforgeeks.org/the-knights-tour-problem-backtracking-1/

https://www.geeksforgeeks.org/warnsdorffs-algorithm-knights-tour-problem/