

Pohl-Warnsdorf's Heuristic

Professor Ira Pohl pohl@cs.ucsc.edu

TA Foaad Khosmood foaad@ucsc.edu

A heuristic is a rule of thumb used to solve a problem. An example might be the shopping heuristic "cheaper prices are found at larger stores."

Hamiltonians in graphs are paths that visit all nodes in the graph -once-and-only once. This is an NP hard problem. A special form of Hamiltonian is a knight's tour.

A knight has an L-shaped move on the chess board. An 8 by 8 chess board can be viewed as a 64 node graph connected by edges represented as a knight's move. So node (1,1) upper left corner has two edges ((1,1),(2,3)) and ((1,1),(3,2)).

Your program written in either Java, Python, C++ or Lisp should attempt to find a Knight's tour using the Pohl-Warnsdorf greedy heuristic.

When at a node - select a neighbor of least degree. (BTW the prerequisites assume you have had basic graph theory - so it is important to review this if you are shaky.) If this value is a tie then look one level deeper and pick a node that has least degree at this second level. If this still leaves ties randomly pick among tied nodes. This homework is due in class April 21, 2009 10am in hardcopy. Late work is not accepted.

Extra-credit: The program can work on a generalized board that is n by m. The program can work on leaper graphs - this is where a knight's move can be altered to be a different l -shape such as a jump of 2 by 3. The program can work on arbitrary graphs. The program runs a comparison with some other heuristic.

requirements

- A working program in a high level language
- Printed output - showing a knight's tour
- one page describing your program and experiments
- tour should be displayed as a sequence of moves that are (x,y) pairs between (1,1) and (8,8).