Finding a Hamilton Path/Circuit.

A Hamilton path of a graph (or digraph) is a path that contains all vertices exactly once. If the last vertex in a Hamilton path has the first vertex as a neighbour, then the path can be turned into a Hamilton circuit by joining the last vertex in the path to the first vertex, i.e. a Hamilton path that is also a cycle is a Hamilton circuit.

Hamilton, Sir William Rowen (1805-1865)
His 'Theory of Systems of Rays' (1827), completed when he was 23, provides a scientific basis for Optics that is still is use today. Hamilton is also the creator of 'Quaternions', a linear algebra of 4-dimensional vectors. The theory of Quaternions was superseeded by the more general theory of Matrices, created by Arthur Cayley (1821-1895). By the age of 13, Hamilton had a mastery of 13 languages, including Hebrew, Sanskrit and Bengali. He entered Trinity in 1823 and before he graduated he accepted the Professorship of Astronomy. Just before the died Hamilton was elected, as the first foreign member, to the National Academy of Sciences of the US.

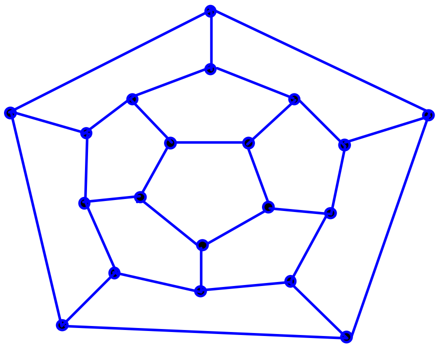
As an application of the Hamilton Cicuit, Hamilton devised a puzzle-game that consisted of a dodecahedron (12 pentagon faces, 'soccer ball'). Each of the 20 vertices (corners) were labelled with the names of cities.

Note: Since the dodecahedron has 20 vertices and 12 faces, it has 30 edges, according to the Euler formula (V+F = E+2).

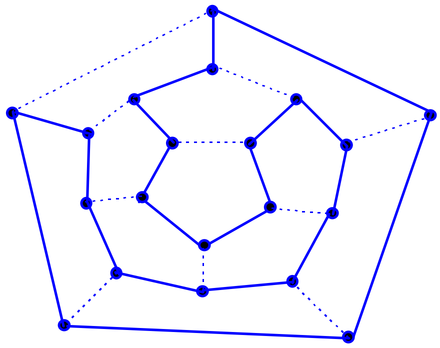
Note: An Euler path / circuit visits each edge exactly once.

The aim of the game was to visit all cities once returning to the start. That is, the aim of the game was to find a Hamilton circuit. We can represent the puzle as a graph.

Find a Hamilton circuit in the following graph.



Solution



For Euler circuits, a graph has an Euler circuit iff every vertex has even degree, but there is no such neat characterisation for a Hamilton circuit.

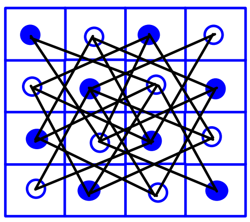
There is no known efficient algorithm for finding a Hamilton path / circuit. In general, finding a Hamilton path / circuit is NP-complete. The fastest known algorithms take exponential time. A particular case of a Hamilton circuit is the Travelling Salesman problem where a salesman wants to visit n cities via the shortest route.

There are simple criteria that are useful in analysing whether a graph has a Hamilton path / circuit.

- If G has a Hamilton circuit, then all vertices have degree ≥ 2.
- If degree of v = 2 then both edges incident on v are in the circuit
- If degree of v > 2 and two edges incident on v are in a Hamilton circuit, then the other edges incident on v are not in the Hamilton circuit.

The Knight's Tour and Hamilton Circuit.

We can model the problem of the Knight's journey / tour by a graph (or digraph) . Each square on the board is a vertext and each possible knight's move is an edge.

a) 

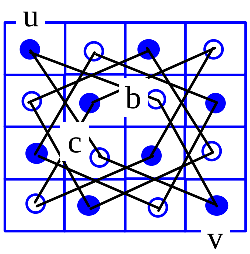
b) 

Diagram a) is the graph representing the moves on a 4x4 board. The 4 'corner' vertices have degree 2, the 8 'side' vertices have degree 3 and the 4 'middle' vertices have degree 4.

On a NxN board, where N is odd, there is no tour (closed journey) as each knight's moves is from a square to an opposite coloured square. Move N^2 (an odd number) will be the same colour as square 1 and so no move is possible from square N^2 to square 1.

There is no tour for a 2x2 board, and since n is odd none for a 3x3 board. We can now show that there is no tour for a 4x4 board. The proof is based on the properties of Hamilton circuits above.

No Knight's Tour on a 4x4 board

Proof:

Consider diagram b), vertex u and v (each fo degree = 2) must be on the Hamilton circuit, if there is one. The edges (u,b) and (b,v) will be in the Hamilton circuit and all other edges from b will not be in the Hamilton circuit. So edges (u,b), (b,v), (v,c) and (c,u) will be on the Hamilton circuit, which is impossible as these edges form a closed circuit. The diagram b) contains four closed circuits. A Hamilton tour is not possible on a 4x4 board.

End proof

It can be proved, by induction, that all NxN boards, with even N, have a Knight's tour (closed journey).

Theorem:

Let G be an (undirected) graph of n vertices. If for each pair of vertices, b and c, we have that the

$$(\text{degree of } b) + (\text{degree of } c) \geq n$$

then G has a Hamilton cycle.

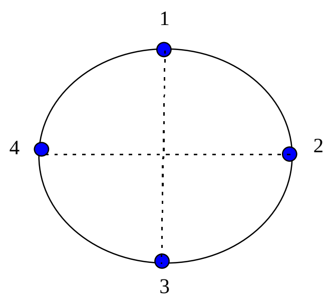
Proof:

For each graph with n vertices the proof is based on induction on the number of edges from m down to 3 (as we need 3 vetices to get a cycle).

Base step:

Every **complete** graph has a Hamilton cycle. In a **complete** graph each vertex has n-1 neighbours and to get a Hamilton cycle put the vertices in a circle and connect them up with a cycle.

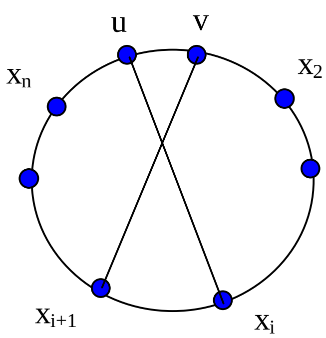
e.g. N= 4



Induction hypothesis:

Assume we can find a Hamilton cycle on graphs, with given condition above, that have $\geq$ m edges.

Find a Hamilton cycle on graph (with conditions above) with m-1 edges. Let G = (V,E) be such a graph (|V| = n, |E| = m-1). Consider any two vertices u and v that are not neighbours. Let G' be the graph that is the same as G except that edge (u,v) is added. By induction, G' has m edges and satisfies the conditions, so there is a Hamilton cycle. This Hamilton cycle may contain edge (u,v), but if not then we are done and we have a Hamilton cycle for G. If the Hamilton cycle for G' does contain the edge (u,v) then we can reconstruct a Hamilton cycle for G out of the one for G'. Without loss of generality (as we can rename nodes) let u = x_1 and v =x_n. Since (degree of u) + (degree of v) ≥ n, there are at least n neighbours of u and v. Besides u and v there are n-2 vertices in the graph G and so there must be 2 neighbouring vertices x_i and x_{i+1} in the Hamilton cycle for G' such that u is a neighbour of x_i and v is a neighbour of x_{i+1}.



Using edges (u,x_i) and (v,x_{i+1}), we have a new Hamilton cycle

u, x_i, .., x_2,v,x_{i+1}, ... x_n, u

in G' that does not use edge (u,v), hence this is a Hamilton cycle in G.

end proof.

Backtracking procedure for finding a Hamilton cycle, if there is one.

In a graph, there may not be Hamilton cycle and the program should indicate this, i.e. the program should not assume that there is a Hamilton cycle.

We need class attributes,

success:BOOLEAN
visited: ARRAY[BOOLEAN] -- visited vertices
hamilton_cycle: ARRAY[INTEGER] -- stores vertices of cycle
G : ARRAY2[BOOLEAN] -- Adjacency matrix for graph

```
find_cycle(j, move : INTEGER) is
    local
        k : INTEGER
    do
        if move = G.count  and then G.item(j,1) then
            success := true
        else
            from
                k := 1
            until
                k > G.count or success
            loop
                if G.item(j,k) and not visited.item(k) then
                    visited.put(true,k)      -- mark k as visited
                    hamilton_cycle.put(k, move+1)
                    find_cycle(k, move+1)
                    visited.put(success, k) --   unmark k,
                                                 if not success
                end
                k := k+1
            end
        end
    end -- find_cycle
```

To find a cycle, use the call,

find_cycle(1,1)

If success then we can print out the Hamilton circuit from the array, hamilton_cycle, otherwise there is no Hamilton circuit.

Note: Before the call of the procedure, find_cycle, we could check for all pairs, b and c, of vertices that (degree of b) + (degree of c) ≥ n but if this is false, there may still be a Hamilton cycle. The theorem say that if

(degree of b) + (degree of c) ≥ n, for all pairs b,c

then there is a Hamilton cycle and so this is a sufficent condition for a Hamilton cycle. If a graph has a Hamilton cycle, this does not mean that all pairs b, c satisfy (degree of b) + (degree of c) ≥ n and so this condition is not a necessary condition for a Hamilton cycle.

If a graph has a Hamilton cycle, then the graph is called a Hamilton graph.