

Hamiltonian Circuits and Algorithm Complexity

A Hamiltonian Circuit is one that visits every vertex exactly once. In most graphs this will not require traversing all the edges. (Contrast this with an Euler circuit where the goal is to traverse every edge once.) Surprisingly there is no known quick way to determine if a graph has a Hamiltonian Circuit. It is an unsolved problem in mathematics. It appears so difficult to do, that mathematicians have tried instead to prove that it can't be solved, so that they can stop trying (there's no shame in not solving an impossible problem, and some fame in proving one is). That too is an unsolved problem! So there are no known good solutions, but we can't stop looking until we can prove there aren't any to be found.

Above I have said there are no known “good” or “quick” ways to determine if there is a Hamiltonian Circuit. What mathematicians mean by that is there is no “polynomial time” algorithm for it. More new terms.

An **algorithm** is a mechanical procedure that is guaranteed to find the correct solution in a finite number of steps (and thus finite amount of time).

Well that's clear, if a little long-winded, and it sounds like having an algorithm is pretty good. The problem is that some finite amounts of time are still *very* large. Let's compare algorithms of different computational orders. Ones where the number of steps to find the answer are proportional to: the natural logarithm of n , i.e. $\ln n$, to n itself, to n^2 , to e^n and to $n!$ Suppose n represents the number of vertices in a graph. For small graphs all the algorithms are manageable:

# vertices	# of steps to solve by different orders of algorithms				
n	$\ln n$	n	n^2	e^n	$n!$
2	1	2	4	7	4
4	1	4	16	55	24
6	2	6	36	403	720
8	2	8	64	2,981	40,320
10	2	10	100	22,026	3,628,800

But even at these small values many of the orders of algorithms are beyond being able to be carried out by hand, and the exponential and factorial might even take a noticeable amount of time on a computer. But look what happens when the problems get just a little larger say 100 vertices:

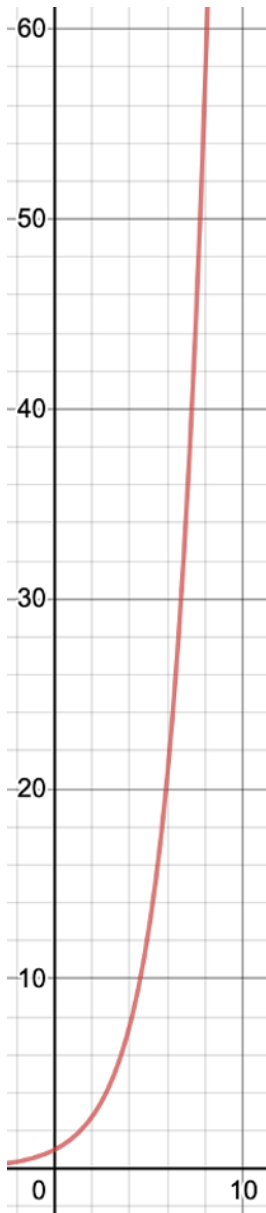
Order of algorithm	# of steps to solve problem
$\ln n$	4.6
n	100
n^2	10,000
e^n	26,881,171,418,161,356,094,253,400,435,962,903,554,686,976
$n!$	93,326,215,443,944,152,681,699,238,856,266,700,490,715,968,264,381,621,468,592,963,895,217,599,993,229,915,608,941,463,976,156,518,286,253,697,920,827,223,758,251,185,210,916,864,000,000,000,000,000,000,000,000
# of atoms in universe	~ 100,000,000,000,000,000,026,609,864,708,367,276,537,402,401,181,200,809,098,131,977,453,489,758,916,313,088

(The estimate for the number of atoms in the universe is provided to give a sense of the scale of the larger numbers.)

Now pause to consider that there are real-world applications of Hamiltonian Circuits and related problems that involve graphs with *millions* of vertices, and you can see why “good” algorithms are needed. So back to good: a starting point for good is a polynomial-time algorithm, i.e. one where the number of steps and thus time is proportional to some polynomial of the size of the problem. Not necessarily as low an order as n^2 , but just **not** an exponential or factorial.

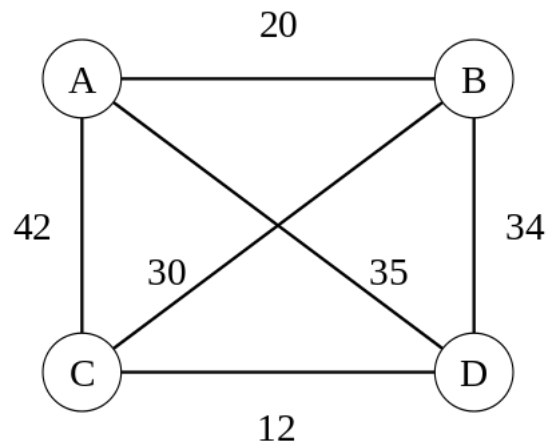
A **polynomial-time** algorithm is one where the number of steps and thus time is proportional to some polynomial of the size of the problem

Algorithms for solving Hamiltonian Circuit problems have progressed from early ones where the number of steps was proportional to $n^2 2^n$ to the current state of the art where the number of steps is proportional to 1.657^n . 1.657 is a small base, but that is still an exponential function and it grows like one:

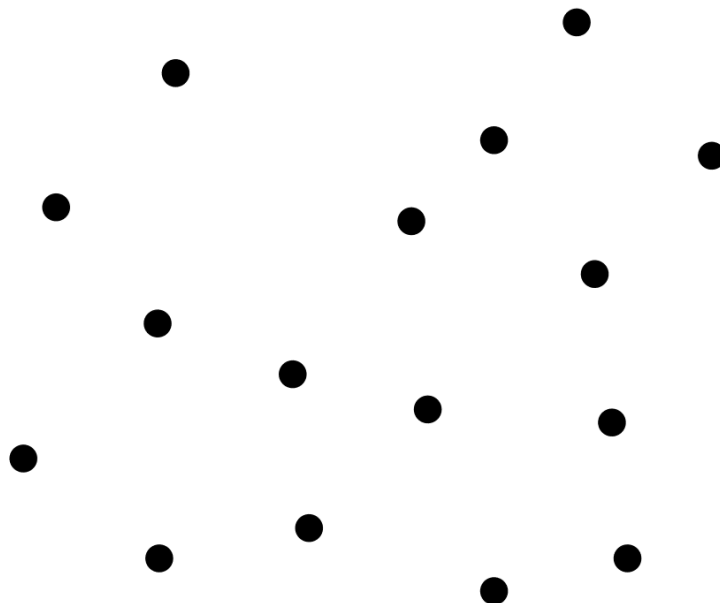


The Problem of finding Hamiltonian Circuits is a special case of the travelling Salesman Problem. “Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?” In Hamiltonian terms this is the problem of finding the Hamiltonian Circuit of Least Weight in a graph.

In a small problem we can see the solution quickly, and verify it by considering all possible solutions and checking that ours is optimal.



But a problem only has to get a little larger before that becomes unworkable:



Since the problem of determining whether a graph has a Hamiltonian circuit has no good solution, it should come as no surprise that neither does the Travelling Salesman Problem. What do we do when we need to solve one then? We rely on approximations and heuristics.

A **heuristic** is a solution procedure that gives a good answer quickly.

A heuristic is **not** guaranteed to find the optimal solution, but it should find a good solution. Some heuristics have clear bounds on how bad their answer might be. Others, don't.

There is a separate document about one very useful (but not graph-theoretic) heuristic called The Rule of 72. Check it out.