

Every graph in graph theory can be defined by two items: the vertices(or nodes) it has, and the edges that connect these vertices. In daily life, we see millions of graphs like these all around us; the tree leaves, lightning storms, even our social media networks all act exactly like really, really large graphs. In fact, graphs can describe really important thing, like protein structure, so well that biologists and chemists use them all the time to look at molecules! The only problem is that when these graphs get very big, it becomes really difficult to tell the difference between two graphs, or even if there is any difference at all!

Luckily, mathematicians and computer scientists have worked for many decades to define this problem, and have simply called it: the Graph Isomorphism Problem. What the Graph Isomorphism Problem states is that it is *really* difficult to tell in any reasonable amount of time whether two graphs are different from each other, or really the same thing. However, all graphs have attributes to them, one of which is their *adjacency matrix*. The concept of an adjacency matrix is simple: it describes the graph exactly by saying which vertices are connected to which. Often, we like to think of an adjacency matrix as the “table form” of a graph, because it’s a lot easier to deal with numbers in a table than dots and lines.

When two graphs are the same, but look different, we call them *isomorphic*. A neat thing about isomorphic graphs is that they really have the same adjacency matrices; the only problem is, one has been shifted around a lot, and we don’t know how it’s been shifted. If we can figure out how that matrix has been shifted, we have essentially solved the Graph Isomorphism problem, but the only problem is, there are so many possible ways that it could have been shifted that it’s practically impossible to try them all out and find the right one.

But what if we could apply some restrictions so that we don’t have to check as many matrices? In a nutshell, that’s what our algorithm does. Given two graphs, we can limit the possible ways that their adjacency matrices were shifted, and we can look through these shifts a whole lot faster as a result.

On average, our algorithm allows for us to check for a lot fewer graphs in order to determine if two graphs are isomorphic or not. Especially for really big graphs, like those that come up in protein and molecule structure, our algorithm can come up with a solution much faster than currently known methods. Unfortunately, our algorithm only really provides an advantage when the graph looks similar to an “average graph” of its size. The good news is, most graphs do! For industrial applications, our algorithm can solve a wide variety of problems that people face due to the Graph Isomorphism Problem.