**To Find A Cycle In A Graph:**

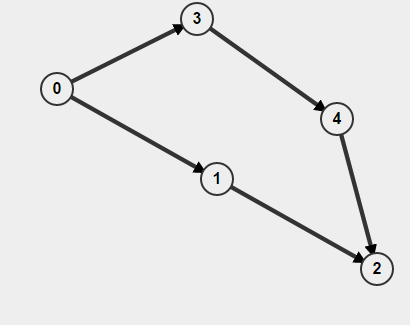
For finding a cycle in a **undirected** graph, either breadth first search or depth first search can be used.

However, for finding a cycle in **directed graph**, only depth first search can be used.

1. **Why can we not use breadth first search for finding cycle?**

Depth first search is more memory efficient than breadth first search as you can backtrack sooner. It is also easier to implement if you use the call stack but this relies on the longest path not overflowing the stack.

Also if your graph is directed then you have to not just remember if you have visited a node or not, but also how you got there. Otherwise you might think you have found a cycle but in reality all you have is two separate paths A->B but that doesn't mean there is a path B->A. For example,



If you do BFS starting from 0, it will detect as cycle is present but actually there is no cycle.

With a depth first search you can mark nodes as visited as you descend and unmark them as you backtrack.