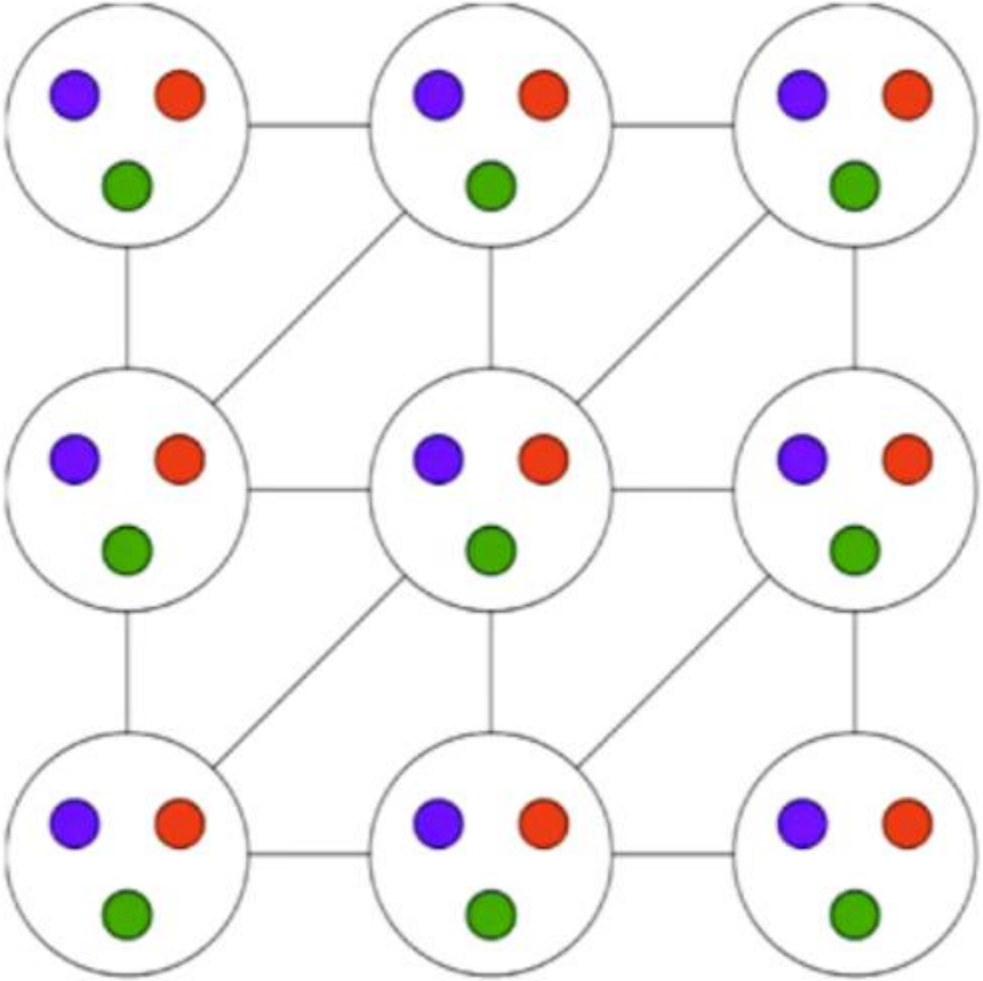
Using Backtracking Search



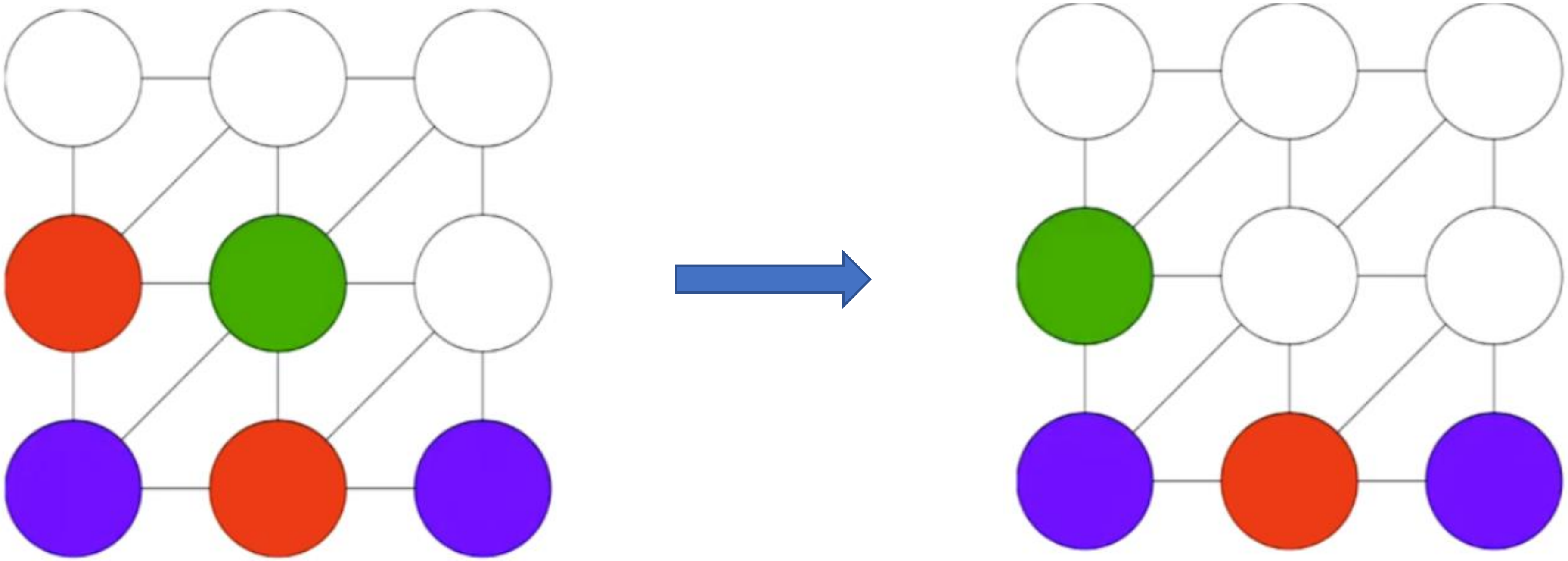
Backtracking Search can be regarded as a Depth-First Search with two constrains:

1. Only one value is considered per step.

2. If no value in the following node’s domain satisfies the constraint in Depth-First Search, there is no need to continue to explore. We call the current node a dead node. In contrast, those nodes that can be explored further are called live nodes. Every time when we meet a dead node, we go back to the last live node and change to another value.

Using Backtracking Search

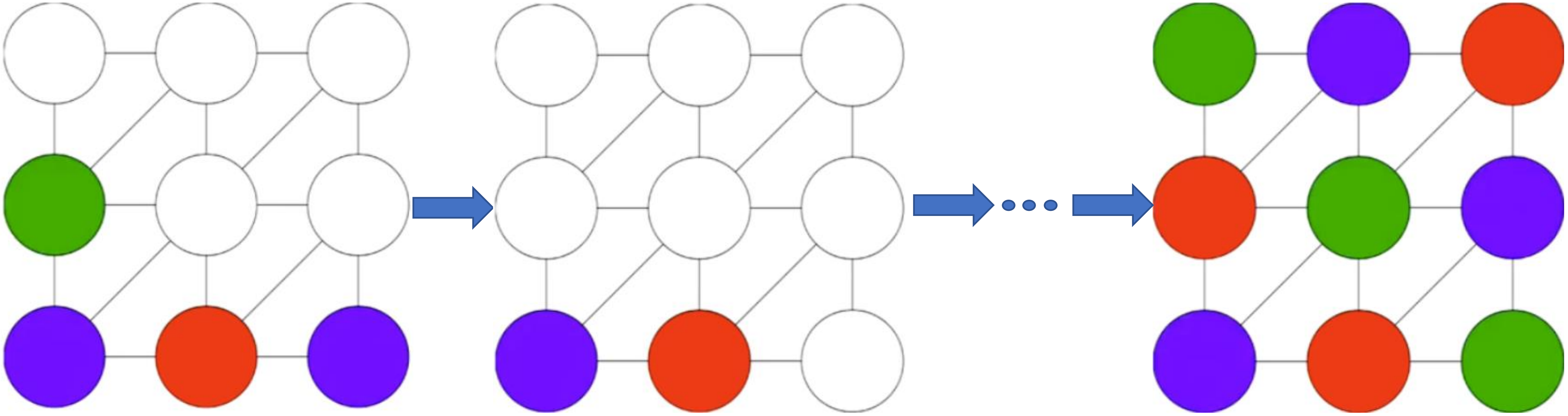
So at the beginning, we paint the nodes according to the rules of depth-first search (remember the tie of colours is broken from purple to red to green). Until the fifth node is painted green, we find that the next node has no colour to satisfy the constraints. Thus, we go back to the last live node and change its colour.



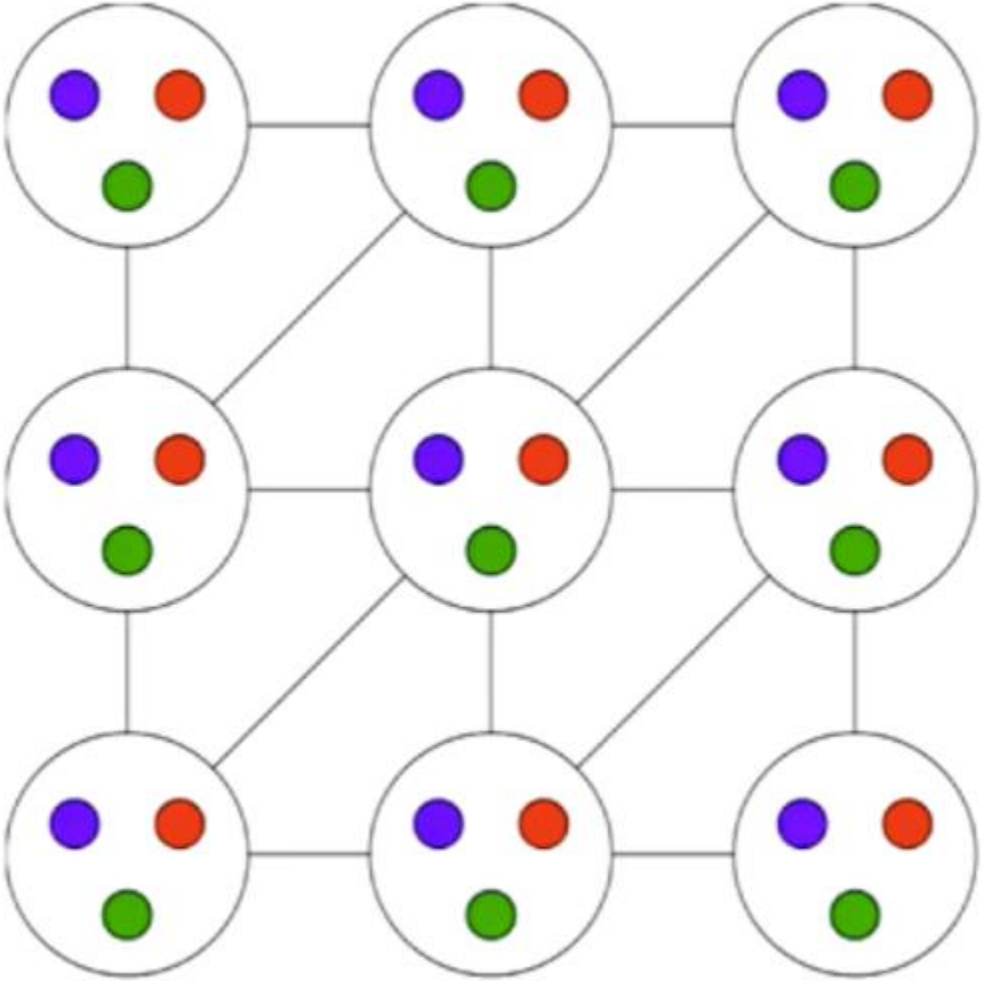
Using Backtracking Search

However, changing the previous node to green, we found that it still doesn't work, so we continue to backtrack.

Follow this way until success:



With Forward Checking and Ordering



Forward Checking:

After colouring the current nodes, the currently selected colour needs to be removed from the domain of all nodes connected to it.

Repeat this step, until the domain in a certain node is empty. It is obvious that the entire colouring task cannot be completed, so it needs Backtracking.

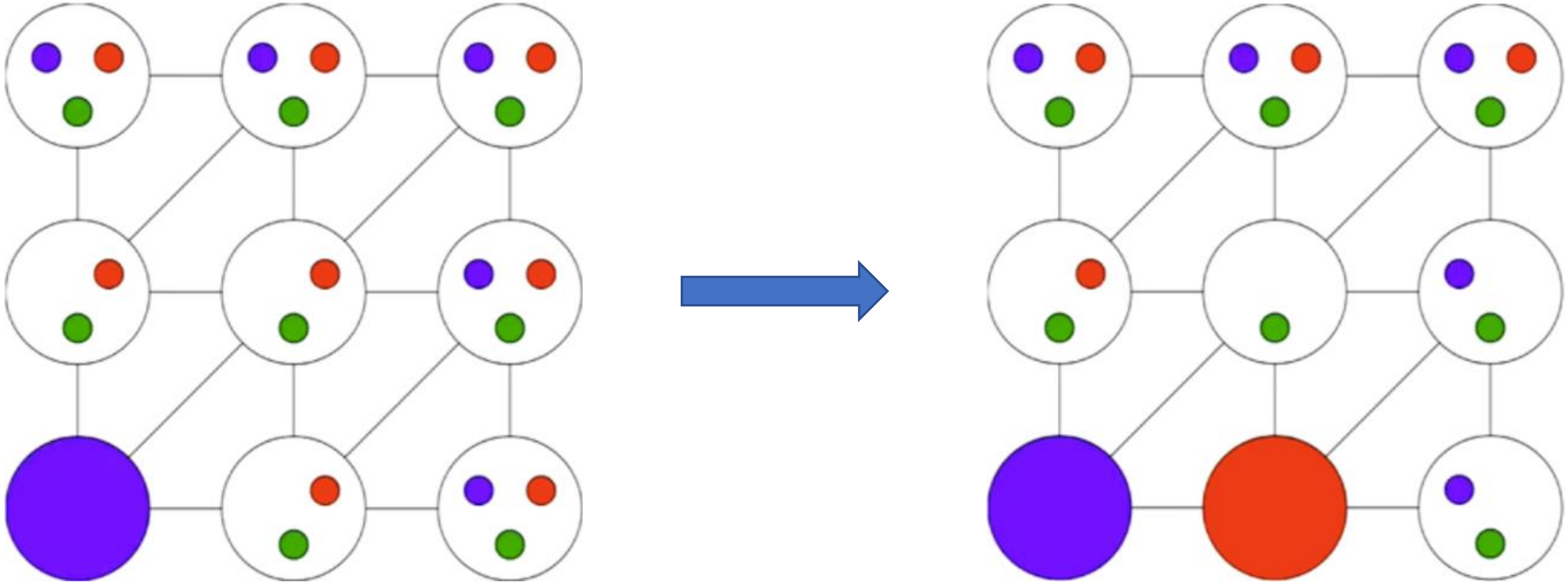
Ordering:

In this case, we firstly consider the nodes with the fewest possible remaining colours.

With Forward Checking and Ordering

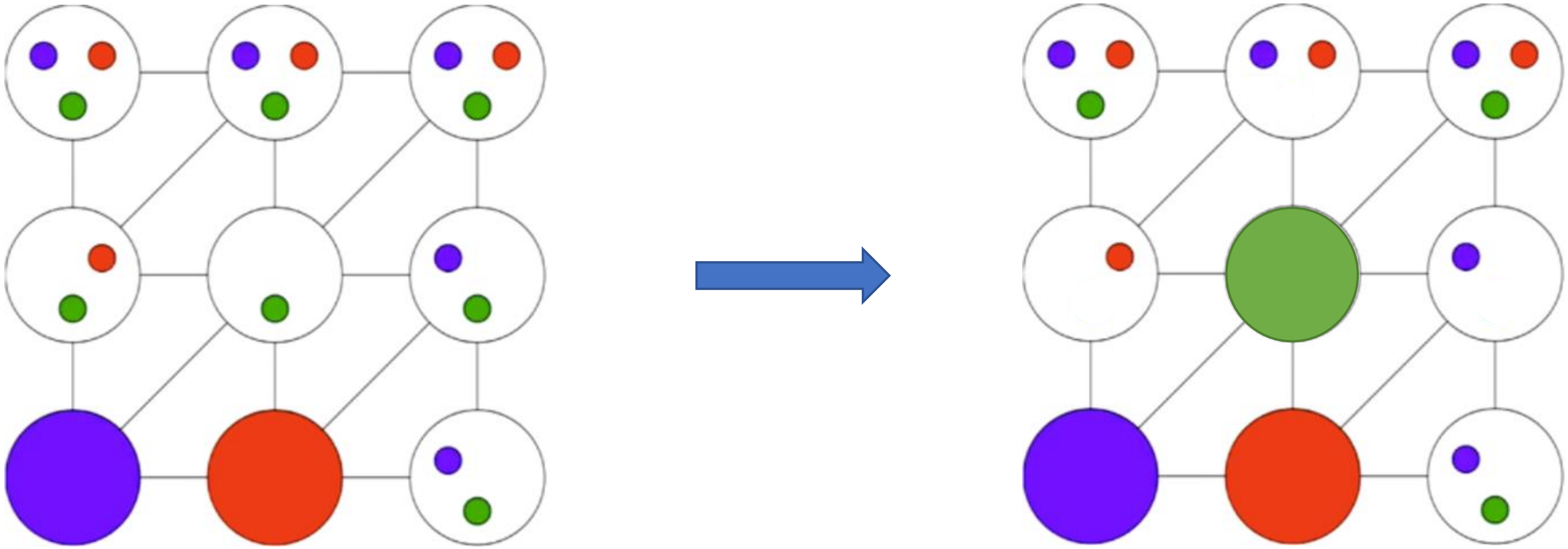
The first node is painted purple, then the purple in the domain of the two adjacent nodes be deleted.

Because the tie of nodes is broken first from left to right and then from bottom to top, and the tie of colours is broken from purple to red to green. The next node is the one on the right and it is painted red.



With Forward Checking and Ordering

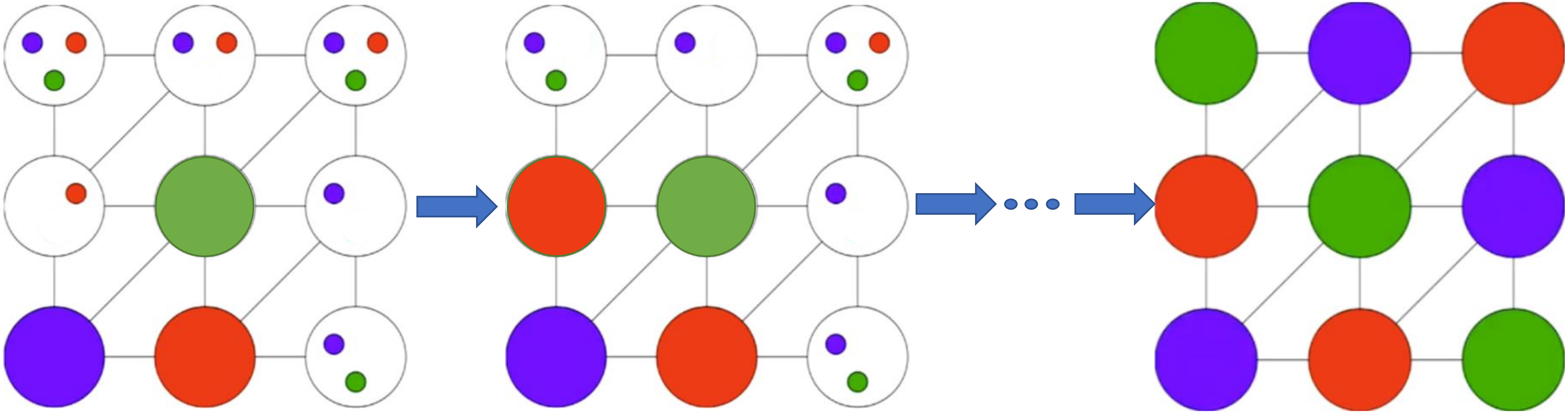
Then the Ordering method works. The node above the red node is chosen because it have fewest colours available. This node is painted green and the corresponding colours of the adjacent nodes are deleted.



With Forward Checking and Ordering

The left and the right nodes of the green one have the same number of available colours. Thus, we now follow the tie breaking rules to select the left node, and it is painted red. Also, please remember to remove the corresponding colours of the adjacent nodes.

Follow this way until success:



Thanks

Hope you have a better understanding of

the Backtracking Search and **the Backtracking Search with Forward Checking and Ordering**