These notes are available at http://web.mit.edu/dxh/www/6034-constraint.odt

6.034 Notes: Four strategies for Constraint Propagation

We typically solve constraint satisfaction problems by drawing a search tree whose nodes contain partial solutions. That way, the task of solving the problem becomes the task of searching a tree—each branch represents a value that we tentatively assign to a variable; we later backtrack if we find that our chosen assignment cannot work.

To make the search more intelligent, we try to prune as many branches as we can from the tree, so that we avoid wasting time with dead ends.

In 6.034, we study four different strategies for pruning branches while you're searching for a solution. In **increasing order of power**, (but also increasing order of the amount of extra work you have to do), the strategies are:

1. Depth first search only.
2. Depth first search + forward checking
3. Depth first search + forward checking + propagation through singleton domains
4. Depth first search + forward checking + propagation through reduced domains

To elaborate:

1. Depth first search only.
This strategy is the most basic approach; **it doesn't prune any branches at all**. The only check it makes is that all the assigned values so far are consistent with each other.

<u>To perform depth first search only:</u>

1. [DFS] After you assign a value to a variable, examine all of the variables you've assigned values so far, and make sure those values are consistent with the constraints. If they aren't, backtrack.

2. Depth first search + forward checking
This strategy eliminates impossible options from neighboring variables.

<u>To perform dfs+forward checking:</u>

1. [DFS] After you assign a value to a variable, examine all of the variables you've assigned values so far, and make sure those values are consistent with the constraints. If they aren't, backtrack.
2. [FC] After you assign a value to a variable, consider all of its neighbors. Eliminate any options from its neighbors that are incompatible with the value you just assigned.

3. Depth first search + forward checking + propagation through singleton domains

This strategy eliminates impossible options from neighboring variables. Then, if any of those neighboring variables have only one option left, it looks ahead to see what else it can eliminate.

1. [DFS] After you assign a value to a variable, examine all of the variables you've assigned values so far, and make sure those values are consistent with the constraints. If they aren't, backtrack.