```
Performance:
    My test case for speedup/2 is to find all the counts and towers for 4X4 puzzles.
    This provides an upper bound as to how long it would take to provide
    I originally wanted to do the same test for a 5X5, but the poor performance of
    my plain_tower/3 when enumerating all 5X5 puzzles made this test not feasible
    with regards to time.
    The ratio of tower/3 to plain_tower/3 to enumerate all 4X4 puzzles is typically
between
    4 and 6.

    For 5 runs on Linux Server 9.
        R = 5.4285714285714288
        R = 5.3928571428571432
        R = 5.3928571428571432
        R = 5.3928571428571432
        R = 5.4285714285714288
    Averaging To a Speedup of:
        5.41

Other Notes on Performance:
    My tower/3 implementation favors powers of 2 beyond a certain threshold.
    For N=32 it can find a result faster than it can find a result for N=24,
    provided on the value of N.

    Additionally, my plain_tower/3 implementation suffers very poor performance
    beyond N=4. For the test cases provided in the spec, when only counts is defined,
    it can take my implementation on the order of a minute of cpu time to find a solution.
    To enumerate all solutions, it can take 5 minutes for N=5.
    It renders the correct solution to the tests in the spec, it simply suffers poor
performance.




Ambiguous Towers
| ?- ambiguous(5,C,T1,T2).

C = counts([4,2,3,2,1],[1,3,2,3,2],[5,3,2,3,1],[1,2,3,2,2])
T1 = [[1,2,3,4,5],[2,1,4,5,3],[4,5,2,3,1],[3,4,5,1,2],[5,3,1,2,4]]
T2 = [[1,2,3,4,5],[3,1,4,5,2],[4,5,2,3,1],[2,4,5,1,3],[5,3,1,2,4]] ?

(8834 ms) yes

/*interestingly, it took only 45 milliseconds to find an ambiguous puzzle for a 6X6*/



Sources:
    Prolog Documentation for Finite Domain
        http://www.gprolog.org/manual/html_node/gprolog054.html

    Standard SWI-Prolog Functions:
        https://github.com/lamby/pkg-swi-prolog/blob/master/library/clp/clpfd.pl
```