

Rocks game as an introduction to dynamic programming – Craig L. Zirbel – 2009

This is a game between two players. It starts with two piles of rocks, ten rocks per pile. On each turn, a player may take one rock from one pile, or two rocks, one from each pile. The player who takes the last rock or rocks wins the game.

1. Set up and play the game with another person. Try to win! Play a few more times. Keep track of whether the first player or the second player wins.
2. Would you rather be the first player or the second player? Why?
3. Whether you are the first player or the second player, can you describe a good strategy for playing the game?

Let's make it simpler. What you were playing can be called the "10-10" game. Try the "1-1" game, with one rock in each pile at the beginning of the game. Does the first player always win? How?

4. Try these games, and write W if the first player can guarantee a win, and L if the first player cannot guarantee a win. You might want to write out a few of the moves along the way.

1-1
0-1
1-0
2-1
2-2
2-0
3-3

5. Let's organize what we are finding out about these "m-n" games. Use the grid below to record whether the first player can guarantee a win (W) or cannot (L), starting with small games and moving up to larger games. Figure out how smaller games help you understand larger games.
6. Now explain the 10-10 game. Can the first player guarantee a win? If so, what should that player do first? If not, what should the second player do?
7. Looking at your analysis of the game, try to describe a strategy for playing that will guarantee a win to the player who follows it.

First	0	1	2	3	4	5	6	7	8	9	10
0	-										
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											

There are two lessons to be learned here. First, if you want to solve a problem that is too large to understand at first, start by solving simpler problems. Second, sometimes you can build up the solution for larger problems directly from the solution to smaller problems. This technique is called *dynamic programming*.