Lone-Divider Method: Example 1

Four players (Abe, Betty, Cory and Dana) are sharing a cake. The cake is divided into four slices x, y, z and w. The table on the right gives the value of each slice in the eyes of each of the players.

	x	y	z	w
Abe	2.83	5.77	3.54	1.86
Betty	4.75	4.75	4.75	4.75
Cory	3.03	2.09	3.26	4.62
Dana	2.21	1.87	2.23	3.69

Step 1: Bidding. What would each player consider a fair share?

First, look for the <u>divider</u> — this is the player who valued all pieces equally, in this case Betty. The divider would accept *any* of the pieces as a fair share (otherwise she didn't do her job properly!)

For the other players (the <u>choosers</u>), examine one at a time — let's take Abe. Determine his overall valuation of the cake by adding up the numbers in his row:

$$2.83 + 5.77 + 3.54 + 1.86 = 14.00.$$

Then divide this number by 4 (the number of players) to determine what Abe would consider a fair share:

$$14.00/4 = 3.50$$

Therefore, Abe would accept any piece that he thinks is worth at least \$3.50. In this case, the fair shares to Abe would be y and z. (Alternatively, Abe's "bid" is $\{y, z\}$.)

Now do the same thing for Cory and Dana. (Note that the first number will typically be different for different players — there's no reason why Cory and Dana should agree on how much the entire cake is worth.) In the following table, the boldface numbers indicate shares that the corresponding player would consider fair.

	x	y	z	w
Abe	2.83	5.77	3.54	1.86
Betty	4.75	4.75	4.75	4.75
Cory	3.03	2.09	3.26	4.62
Dana	2.21	1.87	2.23	3.69

Step 3: Allocation. Is there an allocation of shares that makes each player happy?

This step is a little hard to make systematic; think of it as a puzzle to solve. In general, you should start with players that will only accept one of the shares, and shares that only one player would accept. In this case you could start by assigning share x to Betty since no one else wants it, and by assigning share w to Dana because nothing else will satisfy him. Once you assign a share to a player, cross out the rest of the entries in that row and column of the table:

	x	y	z	w
Abe		5.77^{*}	3.54^{*}	
Betty	4.75^{*}			
Cory		2.09	3.26*	
Dana				3.69*

It is now apparent that assigning y to Abe and z to Cory is the only possibility.

You may discover that it is impossible to assign every player one of the pieces. Then things get more complicated!

Lone-Divider Method: Example 2

Three players (Paul, Quentin and Rosie) are sharing a cake. The cake is divided into three slices x, y, z. The table on the right gives the value of each slice in the eyes of each of the players.

	x	y	z
Paul	\$2	\$1	\$6
Quentin	\$3	\$3	\$3
Rosie	\$1	\$2	\$9

Step 1: Bidding. What would each player consider a fair share?

In this case Quentin is the divider; he would consider any piece a fair share.

Paul thinks the cake is worth \$9 (2+1+6), so a fair share to him would be any piece at worth at least \$3 (=9/3). There's only one piece like this: piece z.

Rosie thinks the cake is worth \$12 (1+2+9), so a fair share to her would be any piece at worth at least \$4 (= 12/3). There's only one piece like this: piece z.

Step 2: Allocation.

Here we have a deadlock where both Paul and Rosie only want piece z. This is the **C-piece** and the other two pieces x, y are the **U-pieces**. (C for "chosen", U for "unchosen").

We resolve the deadlock by giving the divider, Quentin, one of the U-pieces. It doesn't matter which one; let's say Quentin gets x.

We then recombine the other U-piece (y) and the C-piece (z) into a new piece, the **B-piece**. (B for "big.") Paul and Rosie then divide the B-piece using the Divider-Chooser Method — let's say they flip a coin to see which one of them is the divider and which one is the chooser.

Note that the B-piece is worth \$7 (1+6) to Paul and is worth \$10 (1+9) to Rosie. Therefore Paul will get a share of the B-piece worth at least \$3.50 and Rosie will get a share worth at least \$5. These numbers are more than their original fair share numbers, so they are each certainly getting a fair share (and more).