

Let's start filling in the grid.

	1	2	3	4
GUITAR	\$1500 G			
STEREO				
LAPTOP				

Like this, each cell in the grid will contain a list of all the items that fit into the knapsack at that point.

Let's look at the next cell. Here you have a knapsack of capacity 2 lb. Well, the guitar will definitely fit in there!

	1	2	3	4
GUITAR	\$1500 G	\$1500 G		
STEREO				
LAPTOP				

The same for the rest of the cells in this row. Remember, this is the first row, so you have *only* the guitar to choose from. You're pretending that the other two items aren't available to steal right now.

	1	2	3	4
GUITAR	\$1500 G	\$1500 G	\$1500 G	\$1500 G
STEREO				
LAPTOP				

At this point, you're probably confused. *Why* are you doing this for knapsacks with a capacity of 1 lb, 2 lb, and so on, when the problem talks about a 4 lb knapsack? Remember how I told you that dynamic programming starts with a small problem and builds up to the big problem? You're solving subproblems here that will help you to solve the big problem. Read on, and things will become clearer.