I got this query via email:

On homework 4 for the Arkenstone treasure problem, I was wondering how would we know if the path chosen is the greatest one? Would it be the path that leads to the 8th Vault in the 8th Row of 99 gems, similar to the Knapsack Problem that the path that led to the V(n, W) square was the greatest possible path?

And answer with this (adding a bit here, too, toward the end):

First, we are leaning on Dynamic Programming, in part, because a Greedy Algorithm will NOT (in general) solve this problem.

In fact, as near as I can tell I designed this problem so that no matter what you try in a Greedy sense, you'll come up short.  Whether you're greedy from the bottom or greedy from the top -- and there are a couple of ways to be greedy from the top -- you'll end up shy of the max possible.

Bonus hint:  In no row will the Most Precious Path take the largest number of gems.  (I have said too much, but I am proud of having wangled that.  At least I think it's true.)

So you're "stuck" doing it via a Dynamic Programming algo.

Here's a big hint:  pretend the problem ends on row 2.  How would you determine the max possible for two rows?  Once that's done, well, think of that (set of partial) solution a new starting point, and ask what if the problem ended at Row 3?  What's the best I could do there?  And repeat.

But be careful:  I don't mean that the Most Precious Path will pass through the best sum on Row 2 (it might!) but that the sums on Row 2 are your "new" Row 1.  And just as you might or might not (you WON'T) start from the biggest number on Row 1, you might or might not pass through the best possible total on Row 2.

ME

Think about this:

Suppose the board had a 1,000,000 cell at Row 5 column .....

1000 900 800 000 1000000 000 600 1001

1000 900 800 000     000.     000 600 1001

1000 900 800 000     000.     000 600 1001

1000 900 800 000     000.     000 600 1001

1000 900 800 000     000.     000 600 1001

1000 900 800 000     000.     000 600 1001

1000 900 800 000     000.     000 600 1001

1000 900 800 000     000.     000 600 1001

Obviously, you want to end on the million (in this case) because it's better than the rest of the board combined.

But to get there, you'll HAVE to pass through a ZERO.  A Greedy approach would never get you there.

But a Dynamic Programming approach will get you a path of

1000 1000 1000 1000 900 800 000 MILLION

You just have to take it row by row and cell by cell.

And OK, in this case some seem greedy, but not the first!  And if you go up the 1001 side, you miss out by

1005700 to 1005605. (If I have that all right.)

Even in this little toy example.

FUN!

ME

PLUS: Greedy at every stage gets you: 8008…obviously short of the max.

Did I answer the question?

PLUS PLUS: It occurs to me that I did NOT answer the questiom at least not this part:

Would it be the path that leads to the 8th Vault in the 8th Row of 99 gems, similar to the Knapsack Problem that the path that led to the V(n, W) square was the greatest possible path?

The answer here is NO, but kinda.

For the Knapsack, we had a target, and we knew when we got there what the solution to the problem will be.

HERE, we have EIGHT (8) possible targets. There will be eight possible answers when we’re done. We simply take the BEST of those. (And then work backwards—like we did with the PiMonster, perhaps—to construct the Most Precious Path. However, since you are all OOPers, you might build into each square a parent/penultimate/predecessor like we do with Dykstra and be done with it like that. FUN!)