NAME: Please put your name here!

1. Suppose that we are given the following instance of the 0-1 integer knapsack problem.

Item	Weight	Value
1	3	$25
2	2	$25
3	1	$15
4	4	$40
5	5	$50

And the capacity of the knapsack is $W = 6$.

(a) Fill in the "dynamic programming" table defined by the recurrence relation

$$KV[i,j] = \begin{cases} KV[i-1,j], & \text{if } j < w_i \\ \max\{\, KV[i-1,j],\, v_i + KV[i-1, j-w_i]\,\}, & \text{if } j >= w_i \end{cases}$$

where $KV[i,j]$ is the optimal "knapsack value" for a knapsack of capacity j using only items from 1 to i.

Solution:

(b) How many optimal solutions does this instance of the problem have? That is, how many different ways can the knapsack be filled to the optimal value? Explain how to use the table from Part (a) to find all of the optimal solutions.

Solution:

2. The two-dimensional 0-1 integer knapsack problem is defined as follows. We are given a knapsack that has a maximum weight capacity W and also a maximum volume capacity V. We are given n items, and each item has a weight w_i, a volume v_i, and a price p_i (notice that in this problem, v_i is *volume*, not "value"). Maximize

$$\sum_{i=1}^{n} x_i \, p_i$$

subject to the two constraints

$$\sum_{i=1}^{n} x_i\, w_i \le W \qquad \text{and} \qquad \sum_{i=1}^{n} x_i\, v_i \le V$$

where each $x_i \in \{0, 1\}$. (The x_i variables are a fancy way to mathematically say which items are in the knapsack and which items aren't. The i'th item is in the knapsack if $x_i = 1$, and the i'th item isn't in the knapsack if $x_i = 0$). So the knapsack constrains both the weight and volume of what is put in it.

Write down a recurrence relation for a dynamic programming algorithm for this problem. Be sure to include proper initial conditions. Your recurrence relation will be similar to the one in Problem 1. Explain how, and why, your recurrence relation for this problem differs from the one in Problem 1.

Solution:

3. A machine has n components. For each component there are three suppliers. The weight of component i from supplier j is $w_{i,j}$ and its cost is $c_{i,j}$ with $j = 1, 2, 3$. The cost of the machine is the sum of the component costs, and its weight is the sum of the component weights. The problem is to determine from which supplier to buy each component so as to have the lightest machine with cost no more than C.

   Write a recurrence relation for $W[i, j]$, where $W[i, j]$ is the least-weight machine, composed of components 1 through i that costs no more than j. Be sure to include proper initial conditions.

   Solution:

4. A robot can take steps of 1 meter, 2 meters, or 3 meters. We want to find an algorithm that computes the number of ways the robot can walk n meters, with the order of steps taken into account (that is, a 2 meter step followed by a 3 meter step is different than a 3 meter step followed by a 2 meter step).

   (a) Write a recurrence relation for a dynamic programming algorithm. Be sure to include suitable boundary conditions.

   Solution:

   (b) Find the number of ways that the robot can walk 10 meters. Show your "dynamic programming" table.

Solution:

5. A frequent error in written text is the transposition of adjacent letters, as in "witner" instead of "winter". Using two replaces, or one delete and an insert, assigns a cost of 2 to this error. To account for its frequency, we want to assign a cost of 1 to this error. We do this by allowing an additional edit operation:

   Transpose two adjacent letters.

   Rewrite the recurrence relation for computing edit distance to account for this new operation.

   Hint: Look two cells back.

   Solution: