Practice Questions on Dynamic Programming

These questions (and the ones listed under readings) are good preparation for Quiz #3. They are not to be handed in. Some hints follow the questions, but don't look at them unless you are stuck after working on the problem for a while.

If you provide a solution using dynamic programming, then your solution should include the following items.

- A clear definition of the array you are using (including its dimensions and a clear English description of what the value stored in each element of the array is supposed to represent).
- The equations you use to fill in the array, together with some explanation of why the equations are correct.
- Pseudocode that fills in the array.
- Pseudocode that uses the array to print the required output.

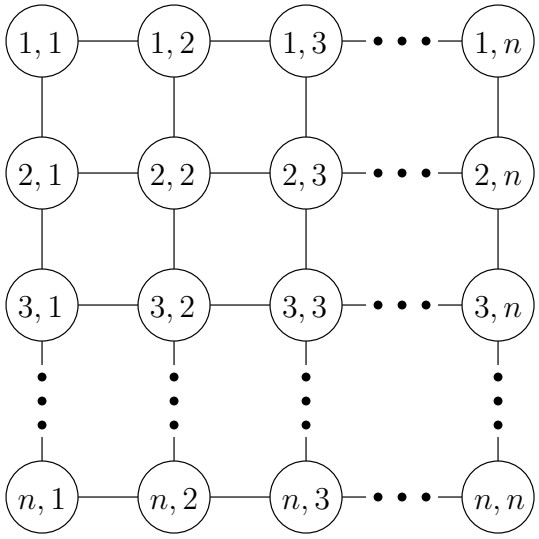
1. Consider the following variation of the knapsack problem. There are n items. Item i has value val_i (in dollars), weight w_i (in kilograms), volume vol_i (in litres). Assume all of these values are integers. Your knapsack can hold items with total weight up to W and total volume up to Vol. Design a dynamic programming solution to find a subset of the n items that will fit into the knapsack and have maximum total value.

2. You are driving a truck from Toronto to Vancouver, a distance of 4500 kilometres. You start in Toronto with a full tank of gas. Your truck's gas tank can hold 200 litres of gas. The truck can travel 1 kilometre on each litre of gas (it has lousy fuel efficiency). There is no constraint on how much gas should be in your tank when you arrive in Vancouver. You have chosen the route you will travel, and you have a map of the n gas stations along the route. You have also found a web site that lists the gas prices at every gas station.

Let d_i be the distance from your starting point to the ith gas station along the route in kilometres and let p_i be the price of gas per litre (in cents) at the ith gas station. (The gas stations are listed in order so that $d_1 < d_2 < \cdots < d_n$ and for all i, $d_{i+1} - d_i \leq 200$.) Each d_i and p_i is an integer. Each time you make a gas purchase (using your debit card), your bank charges you a \$1.50 transaction fee. You may assume that each purchase is an integral number of litres. (Because of the way the problem is set up, there is no advantage to purchasing a fraction of a litre.)

Your goal is to get from Toronto to Vancouver spending as little money on gas as possible. Design an efficient algorithm to print a list describing how many litres of gasoline you should buy at each gas station along the route.

3. Consider an $n \times n$ grid network, as shown below. Each circle represents a processor, labelled by its row and column number. Each edge connecting two processors represents a wire.



The edge connecting processor (i, j) to (i', j') has an associated probability of failure denoted $P(i, j, i', j')$. The processor $(1, 1)$ at the top-left corner of the grid must send a message to the processor (n, n) at the bottom-right corner. It wants to send it along a path with the minimal number of edges. (Thus, the message should always move east or south, never west or north).

Find the path with the minimum probability of failure. (I.e., the path where the probability that the message gets to its destination is maximized.) Assume failures on different edges are independent.

4. Triin and Tiiu are twin sisters who go out trick-or-treating together on Hallowe'en. They agree that when they arrive home afterwards, they will combine all the candy they have received and then split it evenly. Of course, not all types of candy are equally valuable. So the sisters assign a value between 1 and 10 to each piece of candy they receive. Their goal is to split the pile of candy into two piles so that the total value of the first pile is as close as possible to the total value of the second pile. (It may not be possible to split them so that the two piles have *exactly* the same total value.)

When the sisters arrive home, their combined pile of candies contains n items. They enter the values of all the items into an array $A[1..n]$ in some arbitrary order. Then they ask for your help to compute an optimal division of the candies into two piles.

Hints

1. In class, we solved the standard knapsack problem (with a weight constraint, but no volume constraint) using an array M where $M[i, j]$ represented the maximum-value subset of items $i, ..., n$ with total weight at most j. Add an extra dimension to this array to handle the additional volume constraint (so the array you use will be 3-dimensional).

2. Each time you pass a gas station, you have to make a decision: how many litres of gas should you buy there? Your decision will depend upon how much gas you currently have in your tank, so the amount of gas you start with should be one parameter of your subproblems.

3. It is probably better to work with the probabilities that edges *work correctly*. Independence implies that the probability that several edges all work correctly is the product of the probabilities that each edge works correctly. (So if there are 3 edges, each with a 0.1 probability of failing, the probability that all 3 edges work correctly is $0.9 \cdot 0.9 \cdot 0.9 = 0.729$.)

4. Suppose the total value of all the candies is V. The problem is essentially to find whether there are subsets of the n candies whose values add up to some value close to $V/2$. So it is a problem of choosing a subset, like the knapsack problem, but instead of trying to maximize some value of the subset, you just need to know whether the subset exists. So instead of storing values in your solution array, you might just want to store boolean values.