Analysis of Algorithms

Programming Project 3 – Dynamic Programming

In this project, we will consider a variant of the knapsack problem. In this project we are given a set of 320 items, where each item has a price (the amount we must pay to buy it) and a value (how much we value the item – I randomly generated these for this project). The item list and prices are from the game show The Price is Right, where the prices have been rounded up to the nearest dollar. We are given a budget of $100 (you can hard code this), and we want to purchase a set of items that maximizes the sum of the values of the items we picked such that the sum of the prices of the items we pick is at most $100.

Example:

Suppose we have the following items:

- Tickets to a Spurs game, Cost: $100, Value: 80
- Tickets to a concert, Cost: $40, Value: 45
- Tickets to Six Flags, Cost: $30, Value: 40
- Tickets to a comedy show, Cost: $75, Value: 70

Then the optimal solution is to get the concert and Six Flags tickets. This costs $70 total for a value of 85.

In this project, you are to implement a dynamic programming algorithm that computes the optimal solution for this problem for any set of items/costs/values. Again, you can assume that your budget is always going to be $100, but the data set could change from the input provided.

How I suggest you set up your dynamic programming table/subproblems:

- c[i][j] is an n+1 x 101 array, where n is the number of items that we are considering. This should be interpreted as the optmial value such that we are considering only the first i items and we are assuming that we can spend up to j dollars.
- So in the example above, c[2][72] would be the optimal solution if we are only able to consider the Spurs and concert tickets, and we are only able to spend up to $72 dollars. In this case, the optimal value is 45 (we cannot buy the spurs tickets without going over the budget for this subproblem), so we take the concert tickets instead.
- The base case c[i][j] = 0 if i or j is 0 (if you cannot use any items or you cannot spend any money, then clearly the optimal value is 0).
- Then the trick is to figure out how to define c[i][j] as a function of smaller i/j values whenever we have that both i>0 and j>0.

<u>If you do not implement a dynamic programming algorithm for this project, you will get a 0. Unlike the previous project, if you do not use dynamic programming then your solution is very unlikely to be correct for all possible inputs. You can implement your solution top-down or bottom-up, although I'd encourage you to try implementing it bottom-up as this tends to be a bit faster of an algorithm.</u>

How to code this project:

For the most part, you can code this project in any language you want provided that the language is supported on the Fox servers at UTSA. We want you to use a language that you are very comfortable with so that implementation issues do not prevent you from accomplishing the project. That said, we will be compiling and running your code on the Fox servers, so you need to pick a language that is already installed on those machines. See the PDFs on Blackboard in the Programming Projects folder for more information on how to connect to the Fox servers remotely (also covered in the Programming Project 1 overview lecture).

Since everyone is coding in their own preferred language, we are asking you to provide a bash script named *project3.sh* that will act similarly to a makefile. I covered how bash scripts work in class in the Project 1 overview lecture, and I recorded a short follow-up to this here: https://youtu.be/CaIFJWiyU_U

In short, your bash script should contain the command to compile your code, and then on a different line, it should contain the line to execute your code. In this project, we will only be using the items.csv data file, so this file name can be hard coded inside your program. No command line arguments are needed for this project. So, the command to execute your code should look like this:

bash project3.sh

The output should say: "Optimal value: *solutionValue"* on one line (where solutionValue is the sum of the values of the items you are selecting), and then on each line you should print the name of the items that you selected. See the included output file for specifics.

Files provided in the project:

Since you are programming in different languages, we are providing no source files for your code. We have provided a blank project3.sh file for you to fill in as well as the items.csv files. The csv file extension stand for "comma separated values". These files can be opened both by spreadsheets as well as by text editors. The idea is that each line of the file corresponds to a row in the spreadsheet, and the columns are separated by commas. This is a common format when one "column" of data could contains spaces, such as the item names in this project. So you should not read the data in based off of

white space. You should break it up according to commas. This is very easily done in Java using functions like split() (Python has a similar function). It is a little more complicated in C, but here is a nice website showing how it can be done: https://stdin.top/posts/csv-in-c/.

Grading

We will grade according to the rubric provided on Blackboard. The majority of the points come from the following: did the student give a correct implmentation of the selction algorithm that runs in expected O(n) time that returns the correct answers for all possible inputs? Proper documentation may help the grader understand your code and earn you partial credit in the event you have some mistakes in the code.

Violations of the UTSA Student Code of Conduct will be penalized harshly. In particular, be very careful about sending code to a student who asks how you accomplished a particular task. I've heard this story several times recently: "They said they just wanted to see how to perform part X of the project. I didn't think they would submit my exact code." If this happens, you will both be penalized for cheating. To protect yourself and to more properly help your fellow student, send pseudocode, and not actual compilable code.

Also we know about the online sites where people upload projects and have a third party complete the project for you. This is a particularly egregious form of cheating (it's in the best interest of your career to not tolerate this). If you use a solution from one of these sites or submit a minor modification (minor is at the disgression of the instructor) of a solution from one of these sites, you will receive a 0 and will be reported to the university for a viloation of the UTSA Student Code of Conduct.

Submitting

Zip up your project folder and submit on the dropbox on Blackboard by the due date.