ENGSCI 760 | Assignment 3 | Due: 10:00pm, 03/06/2021

Submit all written parts of this assignment into the Canvas assignment dropbox before the due date. Submit your Python code for Q1(f) into the separate dropbox for this.

Question 1: Coin Counting (30 marks)

A central bank in a small, closed economy issues its own coinage in unusual denominations, such as 23¢, 127¢ etc. A shortage of metals has led to a shortage of coins, leading to a government decree that all transactions involving coinage must be done with the minimum number of coins possible.

Suppose that the N denominations of coins in this economy, $D = \{D_1, D_2, ..., D_N\}$, is known. We seek the minimum number of coins that could be used to give a known amount of change x. You may assume that there are sufficient coins available to satisfy the transaction at hand.

(a) Define the stages, states and actions and costs for this problem, assuming that we wish to model this as a dynamic program with backward recursion.

(b) Define mathematically the value function, $V_N(x)$, for all x.
Note that $\frac{x}{D_N}$ is not necessarily an integer. What does this mean and how should this be modelled?

(c) Write down the dynamic programming recursion for this problem.

Suppose that we now wish to code this algorithm up. A colleague has suggested to you that this problem exhibits the properties of 'optimal substructure' and 'overlapping subproblems'.

(d) Explain briefly what this means in the context of this problem.

(e) The colleague has also suggested to you that the deterministic DP approach above is not efficient, and that solving the problem with no explicit recursion may lead to a more efficient algorithm. They also suggest that there is a natural ordering to the problem.

i. Write down a new recursion for $V(x)$ that can be used to solve this problem.
The action set will need to be re-defined.

ii. Suggest what the *natural ordering* might be, and how it makes our formulation more efficient as a result. (Again, think about the actions).

Hint: consider the approaches for finding Fibonacci numbers and rod cutting, covered in Lecture 1.

(f) Hence, or otherwise, code up an algorithm that solves for the optimal number of coins to give a given amount of change. Python must be used to complete this task, as I will be partially auto-marking your function. But this is not a competition; your mark does not depend on other submissions.

The function header is given below. This is available as `coinChange.py` on Canvas.
Read the function definition carefully!

```
def optimalCoinChange(x, denoms):
# Function finds the minimum number of coins required to change a monetary
# amount.
# Inputs:
# x = amount of money to be given in coins, given as an INTEGER, in cents.
#           e.g. $1.35 is input as 135
# denoms = denominations of coins available, in INTEGER cents,
#           given as a ROW VECTOR.
# Output:
# numCoins = optimal number of coins used to find x

# YOUR NAME AND USERNAME GOES HERE
```

Example input / output:

```
x = 35
denoms = [50, 20, 10, 5, 1]

>> optimalCoinChange(x, denoms)
    3
```

Your code will be given a mark for accuracy and efficiency. Your code should be well-documented through comments and/or variable names.

You must submit your Python function file to the Canvas dropbox for Question 1(f). If you do not, you will score 0 for this task.

Please note I will be submitting your code to automated plagiarism checking software.

FYI: the coding task is worth 12 marks, with 3 marks only allocated for efficiency.

Question 2: Exam Beverages (10 marks)

A poor student has just 5 beverages left in his fridge. He wishes to drink these remaining beverages to celebrate the completion of his exams. After each exam, he will consumer either 0, 1 or 2 beverages. The total satisfaction he will receive from the beers is listed in the following table for each of his four exams; a gives the total satisfaction he will gain from consuming one beverage after that exam, and b gives the *total* satisfaction he will gain from consuming two beverages after that exam.

Consuming no beers means that he will gain no satisfaction. Unfortunately, the beverages will expire on the day of his final exam, so any remaining drinks after the final exam cannot be drunk and cannot contribute satisfaction.

Exam	Satisfaction from 1 beer	Satisfaction from 2 beers
Exam 1	20	35
Exam 2	15	37
Exam 3	12	19
Exam 4	9	19

(a) What are the stages, states, and actions for this problem?

(b) What does the value function, $V_n(x)$, represent in this problem?

(c) Write down a dynamic programming recursion, clearly defining any new variables or parameters introduced, and defining the end conditions.

(d) Solve this problem using Excel or your favourite programming language. Include in your submission screenshots from the spreadsheet you made, or the code you used. Note that you do not need to make your code / spreadsheet general – your approach can be tailored for this specific problem instance.

(e) Hence, state in words the student's optimal policy for consuming his five beverages.

Question 3: Candidate Evaluations (10 marks)

Suppose you are interviewing job applicants for a position. There are N applicants to interview lined up outside your office, and you interview them one by one. Assume that each candidate gives a random reward R to your company that is distributed as a random variable with density:

$$f(r) = \begin{cases} \dfrac{6-2r}{9}, & 0 \le r \le 3 \\ \\ 0, & \text{otherwise} \end{cases}$$

After each interview, you discover the realisation of the reward R for the candidate, and may offer them the job, or tell them that you will not hire them. If you do the former, then you stop and get R. If you do the latter, then the candidate is lost. Let V_n be the optimal expected reward after interviewing (and rejecting) n–1 candidates.

(a) Calculate the value of V_N.

(b) What is the optimal decision to make after interviewing and rejecting $N-2$ candidates?

(c) Hence show that $V_{N-1} = 35/27$.

(d) By writing down and evaluating a dynamic programming recursion, show that

$$V_n = 1 + \frac{V_{n+1}^2}{3} - \frac{V_{n+1}^3}{27}$$