CSPB 3104 - Park - Algorithms

Started on	Monday, 26 February 2024, 12:43 PM
State	Finished
Completed on	Monday, 26 February 2024, 12:46 PM
Time taken	2 mins 51 secs
Marks	24.00/24.00
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 11.00 out of 11.00

We are given unlimited coins of the following denominations:

```
[ 1, 5, 20,  50, 75 ]
```

and wish to make up change for a 100 cents.

Suppose we used a simple algorithm that always selects the largest possible denomination (such an algorithm will be called a greedy algorithm):

```
def greedyCoinChange(list of coins, balance)
    select the largest denomination c in list of coins less than or equal to balance.
    newBalance := balance - c
    list := greedyCoinChange(list of coins, newBalance)
    return  (list + [c])
```

How many coins are needed to make change for 100 cents according to the greedy algorithm above?

3

What is the optimal number of coins needed to make 100 cents?

2

Let **numCoins(balance, list of coins)** represent the optimal number of coins for making change for amount **balance** given a **list of coins sorted in descending order**.

What is **numCoins(0,list)?**

0

Let **c** be the largest denomination coin in the list of coins and **balance** be the current balance to make change for.

How many times can the coin **c** be used to make change for **balance**?

- (selected) Some number j between 0 and $\lfloor balance/c \rfloor$ times Correct
- Some number j between 1 and $\lfloor balance/c \rfloor$ times
- Exactly once
- All given choices are incorrect
- exactly $balance/c$ times

Mark 1.00 out of 1.00

The correct answer is: Some number j between 0 and $\lfloor balance/c \rfloor$ times

Going back to our example above with list of coins:

```
[ 1, 5, 20,  50, 75 ]
```

and balance = 100 cents.

What are the number of possible times we can select the 75 cent coin to make change?

- (selected) 0 or 1 times Correct
- We must definitely choose the 75 cent coin
- Choosing the 75 cent coin makes it impossible to make change for the remaining 25 cents

Mark 1.00 out of 1.00

The correct answer is: 0 or 1 times

If we choose 75 cent coin once, what is the optimal number of coins needed to make the balance of 25 cents?

2

If we do not choose the 75 cent coin (i.e, choose it zero times), how many coins do we need ?

- (selected) **numCoins(100, [50, 20,5,1])** Correct
- **numCoins(25, [50, 20, 5, 1])**
- **numCoins (100, [75,50,20,5,1])**

Mark 1.00 out of 1.00

The correct answer is: **numCoins(100, [50, 20,5,1])**

Let the list of coins be $[c_1, c_2, \ldots, c_n]$ in descending order and b be the balance. If $b > c_1$, then fill in the missing parts of the recurrence below:

$$numCoins(b, [c_1, \ldots, c_n]) = ???_1 \begin{cases} ?_2 + numCoins(b, [c_2, \ldots, c_n]) \\ ?_3 + numCoins(b - c_1, [c_2, \ldots, c_n]) \\ \vdots \\ ?_4 + numCoins(b - \lfloor \frac{b}{c_1} \rfloor c_1, [c_2, \ldots, c_n]) \end{cases}$$

Is $???_1$ MIN or MAX? Write down the result in the box below.

MIN

What is $?_2$? Write down in the box below.

0

What is $?_3$? Write down in the box below.

1

Suppose $b = 50$ and $c_1 = 35$, what should be the value of $?_4$? Write down in the box below.

1

Correct

Marks for this submission: 11.00/11.00.

Question **2**

Correct

Mark 6.00 out of 6.00

We have a list of integers $lst[0], lst[1], \cdots, lst[n-1]$ and a lower limit target L.

- We wish to select the least number of elements from lst so that their sum is **greater than or equal to** L.
- As a special case, we are allowed to choose 0 (or no elements) from lst and the sum would be 0. This would work whenever $L \leq 0$, for instance.

Examples:

list is [10, 11, 12, 13, 16, 19, 27]

L = 25

The optimal answer is to choose just one number [27] from the list to make up a sum of 27 which is more than 25.

Same list but with L = -15

The optimal answer is to choose nothing (or 0 numbers) from the list and make up a sum of 0 which is more than -15.

Let **minSubset(i, S)** denote a recurrence that computes the minimum number of integers to be chosen from the sublist $lst[0], \ldots, lst[i-1]$ of the first i elements of the list to achieve a target sum **greater than or equal to** S.

We wish to write a recurrence for **minSubset(i, S)**

Select the correct choice for the base case for **minSubset(i, S) when S < 0?**

- (selected) 0 Correct
- INFINITY
- -INFINITY
- 1

Mark 1.00 out of 1.00

The correct answer is: 0

Select all the correct choices for the base case for **minSubset(i, S) when i = 0, S > 0?**

- 0
- (selected) INFINITY Correct
- -INFINITY
- 1

Mark 1.00 out of 1.00

The correct answer is: INFINITY

Select all the correct choices for the base case for **minSubset(i, S) when S = 0?**

- (selected) 0 Correct, because you are asked to make a sum of 0, that can be done with 0 elements no matter what integers are given to us
- INFINITY
- -INFINITY

- ○ 1

Mark 1.00 out of 1.00

The correct answer is: 0

Consider the recurrence relation for **minSubset(i, S)** partially shown below:

$$minSubset(i, S) : \min \left(\begin{array}{l} ??_1 + minSubset(i-1, S - ??_2) \\ ??_3 + minSubset(i-1, S) \end{array} \right).$$

Select all the correct choices for the missing fields shown by $??_1$.

- ○ 0
- ○ INFINITY
- ◉ 1 Correct, this represents the fact that we use one integer and we have a remaining target of S - lst[i-1]

Mark 1.00 out of 1.00

The correct answer is: 1

Select all the correct choices for the missing fields shown by $??_2$.

- ○ lst[i]
- ◉ lst[i-1] Correct: if we are allowed to use the first i elements, then lst[i-1] is the last element
- ○ 0

Mark 1.00 out of 1.00

The correct answer is: lst[i-1]

Select all the correct choices for the missing fields shown by $??_3$.

- ◉ 0 Correct
- ○ INFINITY
- ○ 1

Mark 1.00 out of 1.00

The correct answer is: 0

Correct

Marks for this submission: 6.00/6.00.

Question **3**

Correct

Mark 7.00 out of 7.00

An investor has B dollars to invest in stocks. There are currently K stocks $s_1, \ldots, s_K$.

Each stock s_i has a price p_i per stock and expected gain of d_i per stock. The investor can buy more than one share in a stock.

The goal is to determine for each stock s_i, how many shares of s_i should the investor buy (call this number n_i), so that

- Invests a maximum of B dollars $n_1 p_1 + \cdots + n_K p_K \leq B$,
- Maximizes the total (expected) gain: $n_1 d_1 + \cdots + n_K d_K$.

Let $G(j, y)$ denote the maximum profit possible while considering the first j stocks $s_1, \ldots, s_j$ and with a budget of y dollars.

Our goal is to calculate $G(K, B)$ considering all K stocks and budget B.

(A) What is the value of $G(j, y)$ when $y = 0$? Write down either a number, PLUS_INF or MINUS_INF in the box below.

0

(B) What is the value of $G(j, y)$ when $y < 0$? Write down either a number, PLUS_INF or MINUS_INF in the box below.

MINUS_INF

(C) What is the value of $G(j, y)$ when $j = 0$? Write down either a number, PLUS_INF or MINUS_INF in the box below.

0

Now we will write a recurrence for G as below:

$$G(j, y) = \max \begin{cases} G(j-1, y), \\ ??_1 + G(j-1, y - p_j), \\ \vdots \\ ??_m + G(j-1, y - m p_j) \end{cases}$$

(D) Select the right term to fill in for $??_1$

- (selected) d_j Correct
- p_j
- 1
- 0
- ∞

Mark 1.00 out of 1.00

The correct answer is: d_j

(E) Select the right term to fill in for $??_m$

- $m \times p_j$

- ○ m
- ◉ $m \times d_j$ Correct
- ○ 0
- ○ $-\infty$

Mark 1.00 out of 1.00

The correct answer is: $m \times d_j$

(F) Select the largest value for m

- ○ $\lfloor y/p_j \rfloor$
- ○ $\lceil y/p_j \rceil$
- ○ ∞
- ◉ $\lfloor y/p_j \rfloor$ Correct
- ○ 1

Mark 1.00 out of 1.00

The correct answer is: $\lfloor y/p_j \rfloor$

(G) What is the smallest value for j for which the recurrence above can be applied?

1

Correct

Marks for this submission: 7.00/7.00.