CS 660 - HOMEWORK III

Q2 $\{S_1, S_2, \ldots S_m\}$ denominations.

If $T[i, j]$ where i total cents and the coin change values $\{S_1 \ldots S_j\}$;

let the total cents amount be 5. and the coin change values be 1, 2, 3.

	1	2	3
1	1	1	1
2	1	2	2
3	1	2	3
4	1	3	4
5	1	3	5

to form this matrix into a dynamic table, we will add a 0^{th} coin value and total cents amount in first row & colm resp.

∴

	0	1	2	3
0	1	1	1	1
1	0	1	1	1
2	0	1	2	2
3	0	1	2	3
4	0	1	3	4
5	0	1	3	5

{total number of change values}

{total cents amounts}

Base cases:

1) For the 0^{th} colm and 0^{th} row i.e. Mat[0][0] is always equal to 1.

→ The rest of the 0^{th} row is always equal to 1

Since there is always 1 way to form those ~~Numbers~~ total amount 0 using 1, 2, 3 i.e. using all of them 0 times.

→ The rest of 0th column will be 0 since we cannot form those total amount with zero coins.

2) Upon noticing the dp matrix, we can logically understand that ~~we can make~~ the no. of ways we can find for 0,..j changes is the number of ways we can find for 0... j-1 changes + no of ways we can find using j as a change.

∴ for e.g. in the table.
~~to~~ get total 3 from 1, 2, 3 coins, we need to first find total 3 from 1, 2 coins and then with the 3rd coin.

if $j \geq 1$: if $i - S[j] >= 0$

$$\therefore \underbrace{Matrix[i][j-1]}_{\text{not considering 3}} + \underbrace{Matrix[i-S[j]][j]}_{\text{consider 3}}$$

3) We repeat this steps from 0 to R+1 {for total rows} and 0 to l+1 {for total table column}

The value in the matrix would be the total no. of ways to find the value i using 0..j changes (coin).

Time complexity for the above problem would be O(l x k)
where l is total table column and row is the total table row.

Pseudo Code

~~NoOfWays (arr, k, l).~~
~~for k in range (k+1) : //Mat = k+1 x l+1.~~
~~Matrix [0][k] = 1~~
~~Matrix [k+1][0] = 0~~
~~for i in range (1,~~

~~for~~

Matrix size = k+1 x l+1.

```
NoOfWays (arr, k, l) :
    for j in range (k+1) :
        Matrix [0][j] = 1
    for i in range (1, l+1) :
        Matrix [i][0] = 0.
    for i in range (1, l+1) :
        for j in range (1, k+1) :
            if j >= 1
                a = Matrix [i][j-1]
            else a = 0.
            if i-s[j] >= 0
                b = Matrix [i-arr[j]][j]
            else b = 0
            Matrix [i][j] = a+b
    return Matrix [k+1, l+1]
```

Q3. For the given example:

let $1+3-2-5+1-6+7$.

Assigning ~~characters~~ index to each characters

	1	+	3	−	2	−	5	+	1	−	6	+	7
indices	0	1	2	3	4	5	6	7	8	9	10	11	12

We can see that for every even index, is associated a number and for every odd index, is associated a sign.

We know that the formulas ~~round~~ for the odd and even numbers are:

$2k$ for even

and $2k+1$ for odd

for all integers.

let us say that we are trying to find the Max between i to j where $i \neq j \in 2k$

∴ If there is a sub parenthesis condition i.e. there is 1 or more numbers between i and j.

We will traverse all the values of k in between i to j such that

$$\text{Max}(i,k) + \text{Max}(k+1, j).$$

But this won't be the case for every numbers with + and − signs.

For Base Cases:

let us consider an example $1 - 6 + 7$.

where $1 = i$ and $7 = j$.

the signs would be ~~equal~~ $2k+1$

Here, Let us say $k = 1$.

$\text{Max}(1, \cancel{k}\,1) - \text{Max}(6, 7)$.

Since the sign is -ve,

→ we do not subtract Max from Max & instead subtract Min from Max.

$\therefore \text{Max}(i, k) - \text{Min}(k+1, j)$. — ①

and if the signs is +ve,

$\text{Max}(i, k) + \text{Max}(k+1, j)$. — ②

→ Now to ~~ge~~ solve the above eqns, we'll need Min fns; and considering +ve & -ve condns for it,

$\text{Min}(i, k) - \text{Max}(k+1, j)$ {for -ve} — ③

$\text{Min}(i, k) + \text{Min}(k+1, j)$ {for +ve} — ④

→ Now if $i = j$; we take array[2i].
for both Max and Min Values.

→ Now for eqns ①, ② + ③, ④ respectively,
we try to find Max of all these eqns from different values of k.
We than take Max of Max of ① + ②
+ take Min of Max of ③ + ④.

Pseudo Code.

\# Since this is a recursion problem, we start from the end.

```
Max ( arr[0.... n] ):
   for (i = n ; i>=0 ; i--) {
MinV[i,i] = MaxV[i,i] = X[2i]

    for (j = i+1 ; i<n ; j++) {

       for (k = i , i<=j-1 ; i++) {
          if X[2j+1] == '+':
               max = MaxV[max, MaxV[i, k]
                       + MaxV[k+1, j]]
               min = MinV[min, MinV[i, k]
                       + minV[k+1, j]]
          else if X[2j+1] == '-':
               max = maxV[Max, MaxV[i, k]
                       + maxV[k+1, j]]
               min = minV[min, minV[i, k]
                       + minV[k+1, j]]
          MaxV[i, j] = max
          Min [i, j] = min.
   return Max[0, n].
```

Forming the dynamic table.

$x[i, i] = x[2i]$

		0	1	2
		1	− 6	+ 7
0	1	1	−5	2
1	−6		6	−13
2	+7			7

i j

Max (1, 7).

Max (10, 1).

− Min (6, 7).

		0	1	2	3	4
		1	−	6	+	7
0	1	1		−8		
1	−					
2	6			6		13
3	+					
4	7					7

6 + 7.

1 − 13

or.

Max (1, 6)

+ Max (7, 7)

1 − 6 = −5

+ 7

= 2

∴ Time complexity is $O(n^3)$ where n is the number of integers in the given expression.

Q4. 14 a) There is definitely a set of coin values for which greedy algo does not always give the smallest possible of coins.

Let us consider coin values $C = [1, 2, 7, 10]$
where $C[0]=1$, $C[1]=2$, $C[2]=7$, $C[3]=10$
To get change for 14.
let $V=14$,
By solving using greedy,
we get 3 coins i.e. 10, 2, 2.

But there is possibility to get coins value using 2 coins i.e. 7, 7.

This can be achieved using dynamic programming.

Q1. a) To show: $1^5 + 2^5 + 3^5 + \dots n^5 = \Theta(n^6)$.

Let us say that $1^5 + 2^5 + 3^5 + \dots n^5 \leq n^6$, for all values of n.

To verify this, let $n=1$.

$1^5 \leq 1^6$, this holds True

Let $n=2$.
$1^5 + 2^5 \leq 2^6$. $\therefore 1 + 32 \leq 64$.

$\cancel{762}$
1024 3125
7776 16807
32768 59049

$\therefore$ $33 \leq 64$, $\therefore$ this holds true.

let $n = 10$.

$1^5 + 2^5 + 3^5 + \dots 10^5 \leq 10^6$

$1 + 32 + 243 + \dots 100000 \leq 1000000$

$220825 \leq 1000000$, this holds true.

Similarly, we can solve the eqn and say that it holds true for all values of n.

let us say that $f(x)$ is an increasing function.

If. $\sum_{k=1}^{n} f(k) \geq \int_0^n f(k)\,dk$.

let $f(k) = k^5$

$\therefore \sum_{k=1}^{n} k^5 \geq \int_0^n k^5\,dk$.

$\sum_{k=1}^{n} k^5 \geq \frac{k^6}{6}$

$6\left(\sum_{k=1}^{n} k^5\right) \geq k^6$

$6(1^5 + 2^5 + \dots n^5) \geq n^6$.

$\therefore n^6 = O(1^5 + 2^5 + 3^5 + \dots + n^5)$

$\therefore 1^5 + 2^5 + 3^5 + \dots n^5 = O(n^6)$

We know that if $f(n) = O(g(n))$ & $g(n) = O(f(n))$ then $f(n) = \Theta(g(n))$

$\therefore 1^5 + 2^5 + 3^5 + \dots n^5 = \Theta(n^6)$.

b) $n^3 + 2n$ is divisible by 3.
Let $f(k) = k^3 + 2k$.

Proof by Induction:
Assuming that $f(k)$ is correct for some positive integer k, i.e. $f(k)$ is divisible by 3.

We can thus say that
$$k^3 + 2k = 3(x) \qquad -(1)$$
where x is any positive integer.

Now substituting k with $k+1$;

$$(k+1)^3 + 2(k+1) \cancel{= 3(x)}$$
$$= k^3 + 3k + 3k^2 + 1 \quad + \quad 2k + 2$$
$$= k^3 + 3k^2 + 5k + 3.$$
$$= k^3 + 2k \quad + \quad 3k^2 + 3k + 3$$
$$= 3x + 3k^2 + 3k + 3$$
$$= 3(x + k^2 + k + 1).$$

We can see that the above eqn is similar to the original eqn (1) and we can deduce that $(x + k^2 + k + 1)$ are some positive integer.

This shows that $n^3 + 2n$ is divisible by 3 for every natural number n.

e.g. $(0) + 2(0) = 3(0)$
$(1) + 2(1) = 3(1)$
$(64) + 2(4) = 3(24)$