CS430 HW8

Junzo Zheng
A20254380

Problem 1.

① recurrence solution

$$\text{coinchange}(V', i, W') = \begin{cases} \text{Coinchang}(V'-d_i, i+1, W'-w_i) & \text{if use } d_i \\ \text{coinchange}(V', i+1, W') & \text{if not use } d_i \end{cases}$$

② Using memory to store ~~possi~~ the state of (v', i, w')
assume $S(v', i, w')$ for all v', i, w' ~~that~~ are null at initial state

```
Coinchang(v', i, w') {
    if s(v', i, w') != null,
        return S(v', i, w')
    else if (di = v' && wi < w')
        S(v', i, w') = true.
        return S(v', i, w')
    else if (i > 0 || v' < 0 || w' <= 0)
        S(v', i, w') = false
        reture S(v', i, w')
    else return coinchange(v'-di, i+1, w'-wi) || coinchange(v', i+1, w')
```

CS430 HW8

Junzle Zhang
A20254389

Problem 2: Assume that we have two strings S_1, S_2, with length n, m, respectively.

The last character of the longest common substring at S_1 is i, at S_2 is j.

$$L[i][j] = \begin{cases} L[i-1][j-1]+1 & \text{if } S_1[i] = S_2[j] \\ 0 & \text{otherwise} \end{cases}$$

~~For~~ for i from 1 to n. → $O(n)$

for j from 1 to m: → $O(m)$

↓

$O(mn)$

if $S_1[i] == S_2[j]$

if $i = 1$ || $j = 1$

$L[i][j] = 1$.

else

$L[i][j] = L[i-1][j-1] + 1$.

else

$L[i][j] = 0$

CS430 HW 8

Junzhe Zheng

A20254389

Problem 3

(a) define $W(i,j)$ is the ~~maximum~~ total weight at (i,j). it is generated by $(i-1,j)$ or $(i,j-1)$

thus

$$W(i,j) = \max\{W(i-1,j), W(i,j-1)\} + c(W(i,j))$$

(b) Using memoized method to solve this problem.

set $W(1,1) = c(W(1,1))$

for i from 2 to n.

$W(i,1) = W(i-1,1) + c(W(i,1))$

for j from 2 to n.

$W(1,j) = W(1,j-1) + c(W(1,j))$

for i from 2 to n

for j from 2 to n:

~~W(i,j)=max{W(j-1) W(j~~

$W(i,j) = \max\{W(i-1,j), W(i,j-1)\} + c(W(i,j))$

return $W(n,n)$