



Tecnológico de Monterrey

Tarea 3: Programación dinámica

Algoritmos Avanzados

David Míreles Gutiérrez A00836010

PseudoCódigo

Python

```
function calculateChange(coins, change):
    numOfCoins = length(coins)
    dp = array of size (change + 1) initialized with INT_MAX
    selectedCoin = array of size (change + 1) initialized with -1

    dp[1] = 0

    for i from 1 to change:
        for j from 1 to numOfCoins:
            if coins[j] <= i and dp[i - coins[j]] != INT_MAX:
                if dp[i] > dp[i - coins[j]] + 1:
                    dp[i] = dp[i - coins[j]] + 1
                    selectedCoin[i] = j

    if dp[change] == INT_MAX:
        return array of size numOfCoins initialized with 0

    result = array of size numOfCoins initialized with 0
    currentValue = change

    while currentValue > 0:
        index = selectedCoin[currentValue]
        result[index] += 1
        currentValue -= coins[index]

    return result
```

Análisis de Complejidad

Python

```
function calculateChange(coins, change):
    numOfCoins = length(coins) # O(1)
    dp = array of size (change + 1) initialized with INT_MAX # O(change)
    selectedCoin = array of size (change + 1) initialized with -1 #
O(change)

    dp[1] = 0 # O(1)
```

```

for i from 1 to change: # O(change)
    for j from 1 to numOfCoins: # O(change * numOfCoins)
        if coins[j] <= i and dp[i - coins[j]] != INT_MAX: # O(1)
            if dp[i] > dp[i - coins[j]] + 1: # O(1)
                dp[i] = dp[i - coins[j]] + 1 # O(1)
                selectedCoin[i] = j # O(1)

if dp[change] == INT_MAX: # O(1)
    return array of size numOfCoins initialized with 0 # O(numOfCoins)

result = array of size numOfCoins initialized with 0 # O(numOfCoins)
currentValue = change # O(1)

while currentValue > 0: # O(change)
    index = selectedCoin[currentValue] # O(1)
    result[index] += 1 # O(1)
    currentValue -= coins[index] # O(1)

return result # O(1)

```

Tomamos en cuenta la complejidad más alta para saber el peor caso, que es **$O(\text{change} * \text{numOfCoins})$** , dado que ocupamos crear la tabla de dp para calcular la cantidad mínima de monedas necesarias para cada posible valor de cambio hasta change. Esto implica que, para cada valor de cambio, debemos revisar todas las monedas disponibles.