

Práctica : Unidad 6 (ADA)

Backtracking

Ejercicio 1 :

dar cambio :

Entrada \rightarrow Cambio (nro) y monedas (array con la cant de monedas)

Salida \rightarrow nro mínimo de monedas para dar cambio

1° Creo un array dp de tamaño 'cambio + 1' y le pongo un valor muy grande

2° $dp[0] = 0$, por si lo que ingresan es cambio de 0, que es 0

3° Voy a iterar cada moneda disponible

voy a ir actualizando dp para todas las cantidades desde el valor de la moneda hasta cambio

en cada i del for

si i puede ser alcanzado usando la moneda actual

$$dp[i] = \min(dp[i], dp[i - \text{valor de la moneda}] + 1)$$

4° devuelvo $dp[\text{cambio}]$

Va a tener una complejidad de $O(n \times m)$ siendo n el valor del cambio y m las monedas.

Conclusión :

Paso 1 : Queremos encontrar el nro mínimo de monedas para alcanzar un valor (cambio), siempre vamos a poder devolver cambio por hay monedas de 1.

Paso 2 : Array dp \rightarrow almacenara el numero mínimo de monedas para alcanzar cada cambio

Paso 3 : Para cada moneda :

- Revisamos cada cantidad de cambio posible
- Vamos actualizando el dp comparando el min entre el valor actual $dp[i]$ con $dp[i - \text{moneda}] + 1$

Paso 4 : Despues de procesar todas las monedas para las cant de cambio $dp[\text{cambio}]$ sera el valor