

147 Dollars

1. Explique brevemente la estrategia de la solución y como hace uso de la estrategia de Memoización.

Para solucionar el problema, se utilizó un arreglo de memoización llamado "arreglo" de tamaño $[valor + 1]$ donde cada posición "subtotal" del arreglo representa la cantidad de combinaciones posibles para obtener dicho subtotal con las monedas y billetes disponibles; el arreglo se inicializa con la primera posición en 1 puesto que la única forma de obtener un subtotal de 0 es no utilizar dinero. Ahora, lo que hace el algoritmo es empezar a recorrer las monedas y billetes disponibles, i.e. la lista "CURRENCIES", y por cada currency va llenando el arreglo de memoización mirando cuantas combinaciones hay registradas para obtener un valor igual a $[subtotal - currency]$ puesto que por cada una de esas combinaciones hay una combinación para obtener el subtotal actual al sumarle currency. De esta forma, el arreglo de memoización se va llenando hasta que se llega al valor final deseado, en cuyo caso se retorna el valor de la posición valor del arreglo la cual contiene la cantidad de combinaciones posibles para obtener el valor deseado. Note que el algoritmo empieza originalmente a llenar el arreglo en el caso de una sola currency, es decir, el algoritmo empieza por solucionar el subproblema de cuantas combinaciones hay para obtener cada subtotal (desde el valor de la currency actual hasta el valor total) con una sola currency y luego con dos, tres, etc. hasta llegar a la cantidad de currencies disponibles, i.e. la cantidad total de billetes y monedas.

2. Su solución es BU o TP.

Partiendo de la pregunta 1 es claro que la solución es BU, puesto que se utilizó un arreglo de memoización para almacenar los resultados de los subproblemas empezando desde el más pequeño, en el cual se tiene un solo billete/moneda, hasta el subproblema más grande, en el cual se tienen todas las monedas y/o billetes disponibles.

3. ¿Cuál es la complejidad espacial y temporal, con una breve explicación?

En primer lugar, la complejidad espacial, para un valor total "n" a analizar, es $O(n)$ debido a que el arreglo de memoización tiene un tamaño de $[n + 1]$. En segundo lugar, se observa que la complejidad temporal del algoritmo, para un valor total "n" a analizar, es $O(n)$ puesto que la cantidad de monedas y billetes, i.e. el tamaño de CURRENCIES (tamaño 11), es constante, por lo que los loops del algoritmo se ejecutan $[11 * n]$ veces, resultando en $O(n)$.