



Departamento de Cs. e Ingeniería de la Computación
Universidad Nacional del Sur



ALGORITMOS Y COMPLEJIDAD

Actividad 4: Tarea de programación N°1

Integrantes:

Brenda Belen Martinez Ocampo. LU: 111106

Matias David Schwerdt. LU: 111667

Planteo del problema

Elon necesita llevar algunos paquetes en su cohete, pero solo puede llevar aquellos cuya suma de pesos esté entre $\lceil W/2 \rceil$ y W , siendo W la capacidad del cohete y $\lceil x \rceil$ la función techo de x . Se pide determinar si es posible, en caso de serlo, encontrar una lista que contenga las posiciones de pesos de los paquetes que cumplan estas condiciones.

Planteo de la estrategia de resolución

Lo hicimos en Python porque tiene una sintaxis sencilla, y porque es un lenguaje de programación de tipado dinámico lo que favorece una resolución centrada en el problema a resolver y no en cuestiones de programación.

El algoritmo propuesto realiza una búsqueda greedy para seleccionar los paquetes que cumplen con las condiciones mencionadas. Primero se establece un límite inferior de $\lceil W/2 \rceil$ y un límite superior de W .

Si el peso de un paquete cumple con la condición, se retorna la lista conteniendo el índice de ese paquete.

Si se llega al límite superior, se detiene la selección.

Si la suma de los paquetes seleccionados está en el rango establecido, se retorna la lista de índices de los paquetes seleccionados, en caso contrario se retorna -1.

Ejemplo representativo que ilustre la solución

Exponemos 3 ejemplos ilustrativos

- Supongamos que Elon tiene un cohete con capacidad $W = 12$ y 7 paquetes con los siguientes pesos: [1, 1, 3, 10, 1, 1, 1]. Para resolver este caso, el algoritmo establece un límite inferior de $\lceil 12/2 \rceil = 6$ y un límite superior de 12. Al encontrar el paquete de peso 10 en la posición 4, se retorna que la solución es llevar ese paquete nada más, indicando que es 1 paquete en la posición 4.
- Supongamos que Elon tiene un cohete con capacidad $W = 10$ y 5 paquetes con los siguientes pesos: [1, 2, 1, 15, 1]. Para resolver este caso, el algoritmo establece un límite inferior de $\lceil 10/2 \rceil = 5$ y un límite superior de 10. El algoritmo selecciona los paquetes con peso entre estos límites y los suma. Por lo tanto, se retorna la cantidad de 4 paquetes y la lista de índices [1, 2, 3, 5]. El índice 4 no se incluye porque el peso de ese paquete sobrepasa la capacidad máxima.
- Supongamos que Elon tiene un cohete con capacidad $W = 2$ y 6 paquetes con los siguientes pesos: [5, 3, 4, 10, 3, 6]. Para resolver este caso, el algoritmo establece un límite inferior de $\lceil 2/2 \rceil = 1$ y un límite superior de 2. Como todos los paquetes pesan más que la capacidad máxima del cohete, Elon no puede llevar ningún paquete y el algoritmo retorna -1.

Input:

3

7 12

1 1 3 10 1 1 1

5 10
1 2 1 15 1
6 2
5 3 4 10 3 6

Output:

1
4
4
1 2 3 5
-1

Explicación del algoritmo que implementa la estrategia

El algoritmo implementa la estrategia mencionada en el planteo de la siguiente manera:

Inicializar el límite inferior y superior, el contador de paquetes y la lista de índices seleccionados.

Recorrer los paquetes y para cada uno:

- Si el peso del paquete está en el rango establecido, retornar la lista con ese paquete.
- Si la suma de los paquetes seleccionados más el peso del paquete es menor o igual al límite superior, se agrega el paquete a la lista de índices seleccionados y se suma su peso al contador de paquetes.
- Si la suma de los paquetes seleccionados es igual al límite superior, se termina el recorrido.

Si la suma de los paquetes seleccionados está en el rango establecido, retornar la lista de índices de los paquetes seleccionados, en caso contrario, retorna -1.

Complejidad temporal y espacial del algoritmo

En el peor de los casos, el algoritmo recorre la lista de paquetes una única vez, y para cada uno realiza operaciones constantes, por lo que su complejidad temporal es $O(n)$. La complejidad espacial depende del número de paquetes seleccionados, pero en el peor caso, cuando se seleccionan todos los paquetes, la lista de índices seleccionados puede tener una longitud de n , lo que implica una complejidad espacial de $O(n)$.

Demostración de correctitud de la estrategia

Para realizar una prueba por inducción del algoritmo que resuelve el problema de Elon, primero es importante mencionar que el algoritmo greedy que realizamos no garantiza encontrar la solución óptima en todos los casos. Realizaremos una prueba por inducción para el algoritmo proporcionado.

Paso 1: Caso base

La inducción se realizará sobre el número de paquetes n . El caso base es cuando $n = 1$. Si el peso del único paquete está en el rango $[W/2, W]$, entonces el algoritmo selecciona correctamente ese paquete. Si no está en ese rango, el algoritmo devuelve -1, lo cual también es correcto.

Paso 2: Hipótesis inductiva

Asumimos que el algoritmo funciona correctamente para $n = k$ paquetes.

Paso 3: Paso inductivo

Debemos demostrar que el algoritmo funciona correctamente para $n = k + 1$ paquetes.

Cuando se agrega el $(k + 1)$ -ésimo paquete, el algoritmo verifica si su peso está en el rango permitido $[\lceil W/2 \rceil, W]$. Si es así, selecciona ese paquete y devuelve la solución. Si no, intenta agregarlo a la suma actual C si no excede el límite superior. Si se puede agregar, actualiza C y seleccionados. Luego, verifica si C está en el rango permitido y devuelve la solución si es así. Si no, devuelve -1.

Dado que el algoritmo verifica si la suma actual C está en el rango permitido antes de devolver la solución, podemos afirmar que el algoritmo encuentra una solución válida para $n = k + 1$ paquetes.

Queda demostrado que el algoritmo que utilizamos para resolver el problema es correcto y siempre encuentra una solución.