

Algoritmos Voraces

Introducción

También conocidos como algoritmos glotones o, en inglés, greedy algorithms, son de los algoritmos más sencillos que existen y se utilizan en problemas de optimización. En los problemas de optimización se requiere minimizar o maximizar algún valor. Dentro del alcance de los algoritmos voraces se incluyen problemas como los siguientes:

- a) El problema de la devuelta, el cual consiste en conformar una devuelta con el menor número posible de monedas.
- b) El problema de encontrar el *árbol de recubrimiento mínimo de un grafo*, por ejemplo, encontrar la forma de unir por carretera varias ciudades, de manera que el costo de esto sea mínimo.
- c) Caminos mínimos. Este problema consiste en encontrar la ruta más corta posible para ir de un lugar a otro.
- d) El problema de la mochila, consistente en encontrar la forma de llenar una mochila de manera que el costo del contenido sea lo máximo posible.
- e) Planificación de tareas. En estos problemas se trata de planificar varias tareas o actividades de manera que se maximice la ganancia obtenida o el número de tareas realizadas.

Qué es un algoritmo voraz

Los algoritmos voraces son usados espontáneamente por las personas en distintas ocasiones, por ejemplo, existe un algoritmo voraz que las personas usan frecuentemente para conformar una devuelta.

Supongamos que un tendero tiene que devolver \$500 y para ello tiene las siguientes 6 monedas:

200 200 200 100 100 100

El tendero tratará de utilizar el menor número posible de monedas y devolverá dos monedas de 200 y una de 100. A nivel subconsciente se ha utilizado un algoritmo voraz que selecciona las monedas de mayor a menor o, más exactamente, en orden no creciente. El funcionamiento de este algoritmo puede detallarse aproximadamente de la siguiente manera:

- a) Se selecciona la primera moneda de 200.
- b) Se acepta esta moneda, porque sirve para conformar la devuelta de \$500. Ya solo resta conformar \$300.
- c) Se selecciona la segunda moneda de 200.
- d) Se acepta esta moneda, porque sirve para conformar los \$300 restantes. Ya solo resta conformar \$100.
- e) Se selecciona la tercera moneda de 200.
- f) Se rechaza esta moneda, porque solo resta conformar \$100.
- g) Se selecciona la primera moneda de 100.
- h) Se acepta esta moneda, porque sirve para conformar los \$100 restantes.
- i) El algoritmo termina, porque las monedas seleccionadas y aceptadas suman \$500 que era lo que había que devolver.

Características de los algoritmos voraces

a) Los algoritmos voraces extraen un subconjunto de un conjunto dado. En el ejemplo, de un conjunto de seis monedas se extrajeron tres:

200 200 200 100 100 100 → 200 200 100

b) Los algoritmos voraces seleccionan elementos de un conjunto en cierto orden. En el ejemplo se seleccionaron las monedas disponibles de mayor a menor.

c) Los algoritmos voraces no reconsideran decisiones. En el ejemplo se observa que la moneda que es aceptada queda definitivamente aceptada, no existe la posibilidad que se rechace posteriormente. Lo mismo en el caso de las monedas rechazadas.

d) Los algoritmos voraces no ensayan alternativas. Si el tendero del ejemplo sigue un algoritmo voraz, no le dice al cliente que espere, que va a ensayar varias formas de conformar la devuelta, para ver cual utiliza menos monedas.

e) La codificación de un algoritmo voraz se caracteriza por tener un bucle, denominado bucle voraz.

Elementos de los algoritmos voraces

A pesar de su sencillez, se aprecian cinco elementos en estos algoritmos:

a) Los candidatos. Son los elementos del conjunto de dado, del cual se seleccionarán elementos. En el ejemplo los candidatos son las monedas. En otros ejemplos podrían ser tareas, archivos de computador, multiplicaciones de matrices, caminos etc.

b) La función solución. Dado un conjunto la función solución dice si este conjunto es una solución del problema o no. La

función solución devolverá VERDADERO si el conjunto es una solución del problema, aunque esta no sea una solución óptima. Veamos algunos ejemplos de valores devueltos por la función solución en el problema de devolver \$500, disponiendo de tres monedas de \$200 y tres de \$100:

solución(200, 200, 100) = VERDADERO
solución(200, 100, 100, 100) = VERDADERO
solución(100, 100) = FALSO

c) La función factible. Esta función indicará si un conjunto dado es subconjunto de alguna solución óptima o no. Veamos algunos ejemplos de valores devueltos por la función factible en el ejemplo de referencia:

factible(200, 200) = VERDADERO
factible(200, 100, 100, 100) = VERDADERO
factible(100, 100) = VERDADERO
factible(200, 200, 100, 100) = FALSO

d) Función de selección. Esta función selecciona los elementos del conjunto dado en algún orden preestablecido. La función de selección va escogiendo el elemento que parezca más prometedor para ser parte de la solución, pero el elemento puede ser posteriormente aceptado o rechazado. En el ejemplo de referencia, la función de selección escogió las monedas en orden no creciente de denominación:

1°	2°	3°	4°
↓	↓	↓	↓
200	200	200	100

e) La función objetivo. Esta función devuelve el valor, que en el momento tiene la variable que se está optimizando. En el ejemplo, la función objetivo devuelve el número de monedas de un conjunto dado. Ejemplo:

objetivo(200, 100, 100) = 3

Nota: En todo algoritmo voraz existen las cuatro funciones mencionadas anteriormente, aunque frecuentemente no están explícitas en la codificación del algoritmo.