GUÍA DE APRENDIZAJE PARA CONCURSANTES ICPC Y IOI: ENCONTRARSE EN EL MEDIO (*MEET IN THE MIDDLE*)

1. Introducción

Digamos que tenemos n números y queremos calcular cuántos subconjuntos tienen la suma de los números x. Esto se puede hacer en tiempo O(2^n) usando la tecnica de máscara bit la cual nos pemite repasar todos los subconjuntos, pero usar la técnica de **encuentro en el medio (*meet in the middle*)** solo toma tiempo O($2^{\frac{n}{2}}$). Esta es una mejora significativa porque el número del exponente se reduce a la mitad.

2. Conocimientos previos

2.1. Máscara de bits

Esta idea algorítmica permite generar todos los posibles subconjuntos de un conjunto de elementos y puede ser utilizados en problemas en donde se debe contar la cantidad de subcojuntos de un conjunto inicial que cumplan con determinadas condiciones. En este caso la máscara serviría para generar todos los posibles subconjuntos y luego bastaría con chequear cada subconjunto generado y contar con aquellos que cumplan con restricciones o condiciones que imponga el problema.

3. Desarrollo

La idea es dividir primero los números en dos conjuntos y revisar los subconjuntos de estos conjuntos, lo que lleva un tiempo O ($2^{n/2}$). En este paso, se calcula la suma de cada subconjunto y se ordenan las sumas. Después de esto, basta con revisar los subconjuntos del primer conjunto y calcular para cada subconjunto de cuántas maneras se puede seleccionar un subconjunto del segundo conjunto para que la suma sea x. Cuando la suma de un subconjunto del primer conjunto es a, se debe elegir del segundo conjunto un subconjunto cuya suma sea $x - a$. Esto se hace de forma eficaz cuando los subconjuntos están en orden.

Como ejemplo, considere un problema en el que se nos da una lista de n números y un número x, y queremos saber si es posible elegir algunos números de la lista para que su suma sea x. Por ejemplo, dada la lista $[2, 4, 5, 9]$ y $x = 15$, podemos elegir los números $[2, 4, 9]$ para obtener $2 + 4 + 9 = 15$. Sin embargo, si $x = 10$ para la misma lista, no es posible formar la suma.

Un algoritmo simple para el problema es revisar todos los subconjuntos de elementos y verificar si la suma de alguno de los subconjuntos es x. El tiempo de ejecución de dicho algoritmo es O(2^n), porque hay 2^n subconjuntos. Sin embargo, utilizando la técnica de encuentro en el medio, podemos lograr un algoritmo de tiempo O($2^{\frac{n}{2}}$) más eficiente. Note que O(2^n) y O ($2^{\frac{n}{2}}$) son de diferente complejidad porque $2^{\frac{n}{2}}$ es igual a $\sqrt{2^n}$.

La idea es dividir la lista en dos listas A y B de modo que ambas listas contengan aproximadamente la mitad de los números. La primera búsqueda genera todos los subconjuntos de A y almacena sus sumas en una lista S_A. En consecuencia, la segunda búsqueda crea una lista S_B a partir de B . Después de esto, basta comprobar si es posible elegir un elemento de S_A y otro elemento de S_B tales que su suma sea x . Esto es posible exactamente cuando hay una manera de formar la suma x usando los números de la lista original.

Por ejemplo, supongamos que la lista es $[2, 4, 5, 9]$ y $x = 15$. Primero, dividimos la lista en $A = [2, 4]$ y $B = [5, 9]$. Después de esto, creamos las listas $S_A = [0, 2, 4, 6]$ y $S_B = [0, 5, 9, 14]$. En este caso, es posible formar la suma $x = 15$, porque S_A contiene la suma 6, S_B contiene la suma 9 y $6 + 9 = 15$. Esto corresponde a la solución $[2, 4, 9]$.

Meet in the middle es una técnica en la que el espacio de búsqueda se divide en dos partes de aproximadamente el mismo tamaño. Se realiza una búsqueda separada para ambas partes y finalmente se combinan los resultados de las búsquedas.

4. Implementación

Para resumir todo el proceso de esta idea:

1. Divida el conjunto dado en dos conjuntos de aproximadamente el mismo tamaño.
2. Calcule las respuestas requeridas para los dos conjuntos formados así y ordene uno del conjunto resultante para prepararlo para la búsqueda binaria.
3. Itere sobre el conjunto de resultados sin ordenar y busque binariamente el valor requerido en el conjunto ordenado.

5. Aplicaciones

La técnica se puede utilizar si existe una forma eficaz de combinar los resultados de las búsquedas. En tal situación, las dos búsquedas pueden requerir menos tiempo que una búsqueda grande. Normalmente, podemos convertir un factor de 2^n en un factor de $2^{\frac{n}{2}}$ usando la técnica del encuentro en el medio.

6. Complejidad

Como ya hemos mencionado anteriormente está tecnica tiene una complejidad O($2^{\frac{n}{2}}$) por si sola por tanto hay que tener en cuenta las operaciones adicionales que se incorpore en la solución de un problema en concreto ya que estas van a aumentar la complejidad de la solución final.

7. Ejercicios

A continuación un grupo de ejercicios que se pueden resolver aplicando esta técnica algorítmica:

- CSES - Meet in the Middle
- LightOJ - Coin Change (IV)
- LightOJ - Funny Knapsack

- CodeChef - Nice subsets
- DMOJ - Coloreando Cadenas