NP-C: Subset Sum

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

.UBA fiuba
FACULTAD DE INGENIERÍA

Subset Sum

Sea

Conjunto de $C=\{w_1,w_2,...,w_n\}$ números naturales

Determinar

Si es existe un subconjunto de C que sume exactamente W

Es un problema de decisión

Relacionado con el problema de la mochila

¿Subset Sum ∈ “P”?

Utilizando programación dinámica

Se puede resolver en tiempo pseudo-polinomial O(Wn)

Si representamos W en bits

El algoritmos crece exponencialmente a la cantidad de bits de W

¿Existe una solución polinomial?

No se conoce!

¿Subset Sum ∈ “NP”?

Dado

C conjunto de n números naturales

W numero a sumar

T certificado con subconjunto de C

Podemos determinar polinomialmente

$$\sum_{t_i \in T} t_i = W$$

Todo $t_i \in C$

$\Rightarrow$ SUBSET SUM $\in$ NP

¿Subset Sum $\in$ "NP-Hard"?

Probaremos que

SUBSET-SUM $\in$ NP-C

Utilizaremos 3DM

3DM $\leq_p$ SUBSET-SUM

3 Dimensional Matching

Dados

3 sets disjuntos X.Y,Z de tamaño n cada uno.

Un set C $\subseteq$ X,Y,Z de triplas ordenadas

Determinar

Si existe un subset de n triplas en C tal que cada elemento de $X \cup Y \cup Z$ sea contenido exactamente en una de esa triplas?

Reducción de 3DM a SUBSET SUM

Sea

I instancia de 3DM

Con

X,Y,Z conjuntos de n elementos

$T \subseteq X \times Y \times Z$ conjuntos de m triplas

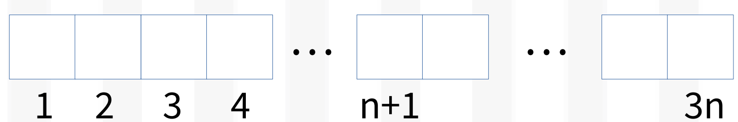
Podemos

Representar cada tripla como un vector de bits

Representación vectorial

Utilizaremos vectores de 3*n bits

- Los primeros n bits representan los elementos del conjunto X
- Los siguientes n bits representan los elementos del conjunto Y
- Los últimos n bits representan los elementos del conjunto Z



A cada elemento de los conjuntos X (y similarmente para Y,Z)

- Les asignaremos un orden arbitrario de 1 a n
- Llamaremos $pos_x(x)$, $pos_y(y)$, $pos_z(z)$ a las funciones que dado el elemento nos retorna su posición en el set

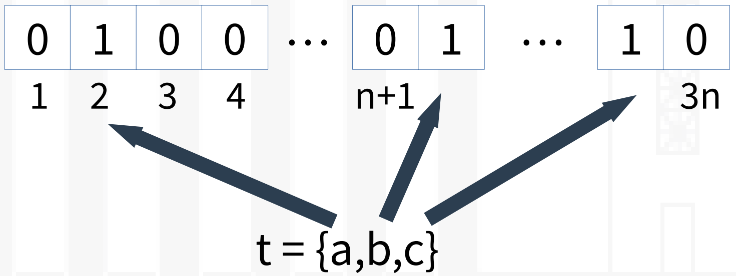
Representación vectorial

A cada t tripla de T

$t = \{x_i, y_j, z_k\}$

La representaremos como un vector V_t de 3n bits

Con todos los bits en cero



Pondremos los siguientes bits en 1

$pos_x(x_i)$

$pos_y(y_j) + n$

$pos_z(z_k) + 2n$

Transformación en un numero entero

Cada vector v_t

Se puede interpretar como un numero w_t en el subconjunto de subset Sum

El Valor W lo formaremos como

el vector con todos los bits en 1

Para sumar W

Tenemos que encontrar aquellos wt que sumados den W

… pero esta propuesta tiene un problema

$$
\begin{array}{r|c|c|c|c|c|c|c|c|c|c|c|}
\cline{2-4}\cline{6-8}\cline{10-12}
 & 0 & 1 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 1 & 0 \\
\cline{2-4}\cline{6-8}\cline{10-12}
 & 0 & 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 1 & 0 & 0 \\
\cline{2-4}\cline{6-8}\cline{10-12}
+ & \vdots & & & & & & & & & & \vdots \\
\cline{2-4}\cline{6-8}\cline{10-12}
 & 1 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 1 \\
\cline{2-4}\cline{6-8}\cline{10-12}
 & \vdots & & & & & & & & & & \vdots \\
\cline{2-4}\cline{6-8}\cline{10-12}
 & 0 & 0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 1 & 0 & 0 \\
\hline
W = & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & 1 & 1 \\
\cline{2-4}\cline{6-8}\cline{10-12}
\end{array}
$$

Problema de Overflow

Puedo elegir 2 o mas números con el mismo bit prendido

Eso genera un overflow

Y podríamos llegar erroneamente a W

$$
\begin{array}{r|c|c|c|c|c|c|c|c|c|c|c|}
 & 0 & 1 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 & 1 \\
 & 0 & 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 1 \\
+ & \vdots & & & & & & & & & & \vdots \\
 & 1 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 1 \\
 & \vdots & & & & & & & & & & \vdots \\
 & 0 & 0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 1 & 0 & 0 \\
\hline
W = & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & 1 & 1
\end{array}
$$

Para evitarlo

Podemos redefinir como expresar wi en base al vector

Transformación en un numero entero

Seleccionaremos una base d

Y representaremos cada tripla como un número de la manera:

$$w_t = \sum_{i=1}^{3n} v_t[i] * d^{i-1}$$

Como todos los elementos del vectores estarán en 0 excepto 3 se pude ver como:

$$w_t = d^{pos_x(x_i)-1} + d^{pos_x(y_i)+n-1} + d^{pos_x(z_i)+2n-1}$$

Para la base d

Utilizaremos un valor mayor a m (cantidad de triplas) → d = m+1

(con eso, aunque las m triplas contengan un mismo elemento no será posible el overflow)

Ejemplo

Sea la instancia de 3DM

$X = \{x_1, x_2, x_3\}$

$Y = \{y_1, y_2, y_3\}$

$Z = \{z_1, z_2, z_3\}$

$T = \{ (x_1,y_1,z_1) , (x_2,y_2,z_3), (x_3,y_3,z_1), (x_1,y_1,z_2), (x_3,y_3,z_2), (x_1,y_3,z_2) , (x_3,y_3,z_3) \}$

Tenemos

$n=3$

$m=7$

Ejemplo (cont.)

Definimos

- Base a utilizar: d=m+1=8
- Tamaño del vector: 3*n=9

Calculamos

- El vector de cada tripla vt
- El vector W

Representamos

- Cada vector como un numero en base d

t	v_t		w_t
(x_1,y_1,z_1)	001001001	262144+512+1	262657
(x_2,y_2,z_3)	100010010	16777216+4096+8	16781320
(x_3,y_3,z_1)	001100100	262144+32768+64	294976
(x_1,y_1,z_2)	010001001	2097152+512+1	2097665
(x_3,y_3,z_2)	010100100	2097152+32768+64	2129984
(x_1,y_3,z_2)	010100001	2097152+32768+1	2129921
(x_3,y_3,z_3)	100100100	16777216+32768+64	16810048
W	111111111		19173961

Ejemplo (cont.)

La instancia de Subset Sum

C = {262657, 16781320, 294976, 2097665, 2129984, 2129921, 16810048}

W=19173961

Resolvemos

(en este caso por fuerza bruta... es NP-C)

Y verificamos que triplas conforman el 3DM

$$
\begin{array}{rrl}
 & 262657 & \rightarrow (x_1,y_1,z_1) \\
+ & 16781320 & \rightarrow (x_2,y_2,z_3) \\
 & 2129984 & \rightarrow (x_3,y_3,z_2) \\
\hline
 & 19173961 & =W
\end{array}
$$

SUBSET SUM es NP-C

Como

SUBSET SUM $\in$ NP

Y 3DM $\leq_p$ SUBSET SUM

Entonces

SUBSET SUM $\in$ NP-C

Presentación realizada en Junio de 2020