

Algoritmia y Complejidad

Ejercicios del tema 7 (Sipser)

Curso 2021 - 2022

UNARY-SSUM

Ejercicio 7.44.

Autor de la presentación:

❖ Marcos Hidalgo Baños

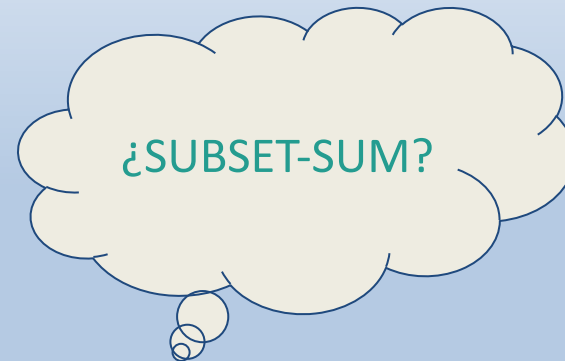
Duración estimada:

❖ 7 minutos aprox.


Enunciado del problema

Sea UNARY-SSUM el problema de la **suma de subconjuntos** en el cual todos los números son representados en **unario**.

- *¿Por qué la demostración NP-Completo para SUBSET-SUM falla para demostrar que UNARY-SSUM es NP-Completo?*
- *Prueba que UNARY-SSUM $\in P$.*



Definición de conceptos

- **Sistema numérico unario.** 
 - Es el sistema de representación numérica más sencillo. Consiste en representar cualquier número natural N mediante una **cadena de unos** de longitud N.

<i>Unario</i>		1	11	...	11111
<i>Natural</i>	0	1	2	...	5

- **Problema de la suma de subconjuntos.**
 - Dado un conjunto no vacío de enteros S, ¿existe algún subconjunto cuya **suma sea otro número objetivo t**?

$S = \{-7, -3, -2, 5, 8\} \rightarrow$ Tiene solución para $t = 0$ ya que $5 + (-2) + (-3) = 0$

SUBSET-SUM \in NP-Completo

- SUBSET-SUM \in NP

Dado una instancia del problema y su solución (certificado) podemos **verificar** en tiempo polinómico si es correcta o no.

También podemos construir M **mTnD** que realice la misma comprobación (demostración alternativa).

M = "Con la entrada $\langle S, t \rangle$

1. De forma no determinista seleccionar un subconjunto c de los números de S .
2. Comprobar si c es una colección de números que suma t .
3. Si es así, aceptar; sino, rechazar."

SUBSET-SUM \in NP-Completo

- 3SAT \leq_p SUBSET-SUM

3SAT = { $\langle \phi \rangle$ / ϕ es una 3CNF fórmula booleana satisfacible}

Tabla de dimensiones $T_{(k+l) \times (k+l)}$

Nº de pasos = $(k+l)^2 \in O(n^2)$

Conclusión

Hay que asegurarse que la reducción se puede realizar en **tiempo polinómico**.

Variables Cláusulas

	1	2	3	4	...	l	c_1	c_2	...	c_k
y_1	1	0	0	0	...	0	1	0	...	0
z_1	1	0	0	0	...	0	0	0	...	0
y_2		1	0	0	...	0	0	1	...	0
z_2		1	0	0	...	0	1	0	...	0
y_3			1	0	...	0	1	1	...	0
z_3			1	0	...	0	0	0	...	1
\vdots					\ddots	\vdots	\vdots		\vdots	\vdots
y_l						1	0	0	...	0
z_l						1	0	0	...	0
g_1							1	0	...	0
h_1							1	0	...	0
g_2								1	...	0
h_2								1	...	0
\vdots									\ddots	\vdots
g_k										1
h_k										1
t	1	1	1	1	...	1	3	3	...	3

S

Volviendo a las preguntas...

→ *¿Por qué la demostración NP-Completo para SUBSET-SUM falla para demostrar que UNARY-SSUM es NP-Completo?*

En SUBSET-SUM podemos representar los elementos del conjunto S en **espacio logarítmico** gracias a que están representados en binario o cualquier otra base mayor.

- *Se necesita de espacio $\log(n)$ para representar el valor n .*

Si la entrada es de tamaño N , su representación en unario es 2^N . Como este **crecimiento es exponencial**, la demostración de que $3SAT \leq_p UNARY-SSUM$ no nos sirve porque la Máquina de Turing requiere de un tiempo mayor que el polinómico.

UNARY-SSUM \in P

Idea general: Como sabemos que SUBSET-SUM \in NP, podemos determinar que UNARY-SSUM \in P si transformamos el problema para que sea un **caso particular** de suma de conjuntos en **binario**.

Entradas:

- Conjunto de números naturales en unario, S.
- Valor objetivo representado en unario, t.

Transformación **unario-binario** para cada uno de estos valores:

Sucesión de divisiones enteras \rightarrow ¡Tiempo polinómico!

Tamaño de la entrada reducido a $O(\log n)$

Funcionamiento:

Simular la mTnD en una mTD sobre la entrada en binario.

$O(2^{\log n}) = O(n) \rightarrow$ ¡También en tiempo polinómico!

Referencias bibliográficas

Michael Sipser, Introduction to the Theory of Computation, Cengage Learning, 2013 (International edition, 3rd edition).

Demostración NPC para el problema de la suma de subconjuntos.
<https://www.geeksforgeeks.org/subset-sum-is-np-complete/#:~:text=Subset%20Sum%20is%20NP%2DHard,i%2C%20ai%3Di.>

Demostración UNARY-SSUM pertenece a la clase de complejidad P.
<http://www.cs.virginia.edu/~evans/cs3102-s10/ps/ps6/ps6-comments.pdf>

