Complexidade computacional

Classifica os problemas em relação à dificuldade de resolvê-los algoritmicamente.

CLRS 34

Lista 8: dúvidas

Exercício 6: Escreva um algoritmo que encontre um arco cuja remoção causa o maior aumento na distância de um vértice s a um vértice t.

Exercício 10: Seja $G = (V, E)$ um digrafo com pesos $w : E \to \{0, 1, \ldots, W\}$ para algum W. Modifique o algoritmo de Dijkstra para que compute os caminhos mínimos a partir de um vértice s em tempo $\mathrm{O}((|V| + |E|) \lg W)$.
(*Dica:* Quantas estimativas distintas de caminhos mínimos podem existir em $V - S$ em cada iteração do algoritmo?)

Cobertura por vértices

Um conjunto S de vértices de um grafo G é uma cobertura se toda aresta de G tem uma ponta em S.

Problema: Dado um grafo G e um inteiro k,
G possui uma cobertura com $\leq k$ vértices?

Cobertura por vértices

Um conjunto S de vértices de um grafo G é uma cobertura se toda aresta de G tem uma ponta em S.

Problema: Dado um grafo G e um inteiro k, G possui uma cobertura com $\leq k$ vértices?

Você consegue provar que este problema é NP-completo?

Alguém fez?

3-Coloração

Uma 3-coloração de um grafo $G = (V, E)$ é uma função $c : V \to \{1, 2, 3\}$ tal que $c(u) \neq c(v)$ para toda aresta $uu \in E$.

Um grafo G é 3-colorível se existe uma 3-coloração de G.

3-Coloração

Uma 3-coloração de um grafo $G = (V, E)$ é uma função $c : V \to \{1, 2, 3\}$ tal que $c(u) \neq c(v)$ para toda aresta $uu \in E$.

Um grafo G é 3-colorível se existe uma 3-coloração de G.

Problema: Dado um grafo G,
G é 3-colorível?

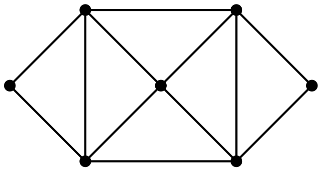
3-Coloração

Uma 3-coloração de um grafo $G = (V, E)$ é uma função $c : V \to \{1, 2, 3\}$ tal que $c(u) \neq c(v)$ para toda aresta $uu \in E$.

Um grafo G é 3-colorível se existe uma 3-coloração de G.

Problema: Dado um grafo G,
G é 3-colorível?

Exemplo:



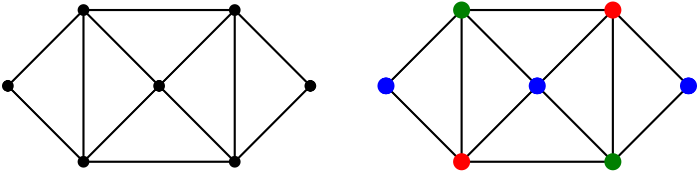
3-Coloração

Uma 3-coloração de um grafo $G = (V, E)$ é uma função $c : V \to \{1, 2, 3\}$ tal que $c(u) \neq c(v)$ para toda aresta $uu \in E$.

Um grafo G é 3-colorível se existe uma 3-coloração de G.

Problema: Dado um grafo G,
G é 3-colorível?

Exemplo:



3-Coloração

Uma 3-coloração de um grafo $G = (V, E)$ é uma função $c : V \rightarrow \{1, 2, 3\}$ tal que $c(u) \neq c(v)$ para toda aresta $uu \in E$.

Um grafo G é 3-colorível se existe uma 3-coloração de G.

Problema: Dado um grafo G,
G é 3-colorível?

Consegue mostrar que 3-coloração está em NP?

3-Coloração

Uma 3-coloração de um grafo $G = (V, E)$ é uma função $c : V \to \{1, 2, 3\}$ tal que $c(u) \neq c(v)$ para toda aresta $uu \in E$.

Um grafo G é 3-colorível se existe uma 3-coloração de G.

Problema: Dado um grafo G,
G é 3-colorível?

Consegue mostrar que 3-coloração está em NP?

Consegue provar que 3-coloração é NP-completo?

3-Satisfatibilidade

Problema: Dada uma fórmula booleana ϕ nas variáveis $x_1, \ldots, x_n$ em que cada cláusula tem exatamente 3 literais, existe uma atribuição

$$t : \{x_1, \ldots, x_n\} \rightarrow \{\text{VERDADE}, \text{FALSO}\}$$

que torna ϕ verdadeira?

3-Satisfatibilidade

Problema: Dada uma fórmula booleana ϕ nas variáveis $x_1, \ldots, x_n$ em que cada cláusula tem exatamente 3 literais, existe uma atribuição

$$t : \{x_1, \ldots, x_n\} \to \{\text{VERDADE}, \text{FALSO}\}$$

que torna ϕ verdadeira?

Exemplo:

$$\phi = (x_1 \vee \bar{x}_1 \vee \bar{x}_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4)$$

Exemplo 4

3-Satisfatibilidade $\prec_P$ 3-Coloração

Exemplo 4

3-Satisfatibilidade $\prec_P$ 3-Coloração

Descreveremos um algoritmo polinomial que recebe uma fórmula booleana ϕ com exatamente 3 literais por cláusula, e devolve um grafo G tal que

$$\phi \text{ é satisfatível} \Leftrightarrow G \text{ é 3-colorível.}$$

Exemplo 4

3-Satisfatibilidade $\prec_P$ 3-Coloração

Descreveremos um algoritmo polinomial que recebe uma fórmula booleana ϕ com exatamente 3 literais por cláusula, e devolve um grafo G tal que

$$\phi \text{ é satisfatível} \Leftrightarrow G \text{ é 3-colorível.}$$

O conjunto de vértices de G contém $\{x_1, \ldots, x_n\}$ e $\{\bar{x}_1, \ldots, \bar{x}_n\}$. Além disso, contém três vértices especiais, TRUE, FALSE, RED, e cinco novos vértices por cláusula de ϕ.

Exemplo 4

3-Satisfatibilidade $\prec_P$ 3-Coloração

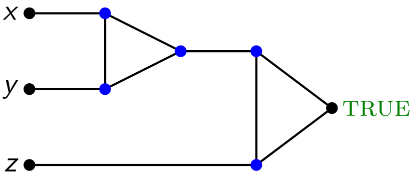
O conjunto de vértices de G contém $\{x_1, \ldots, x_n\}$ e $\{\bar{x}_1, \ldots, \bar{x}_n\}$. Além disso, contém três vértices especiais, TRUE, FALSE, RED, e cinco novos vértices por cláusula de ϕ.

Exemplo 4

3-Satisfatibilidade $\prec_P$ 3-Coloração

O conjunto de vértices de G contém $\{x_1, \ldots, x_n\}$ e $\{\bar{x}_1, \ldots, \bar{x}_n\}$. Além disso, contém três vértices especiais, TRUE, FALSE, RED, e cinco novos vértices por cláusula de ϕ.

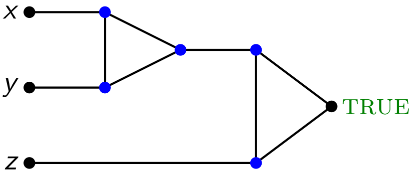
Gadget com cinco novos vértices para a cláusula $(x \vee y \vee z)$:



3-Satisfatibilidade $\prec_P$ 3-Coloração

O conjunto de vértices de G contém $\{x_1, \ldots, x_n\}$ e $\{\bar{x}_1, \ldots, \bar{x}_n\}$. Além disso, contém três vértices especiais, TRUE, FALSE, RED, e cinco novos vértices por cláusula de ϕ.

Gadget com cinco novos vértices para a cláusula $(x \vee y \vee z)$:



Além destas arestas, temos um triângulo nos vértices especiais, e um triângulo em x_i, $\bar{x}_i$ e o vértice RED, para todo i.

3-Satisfatibilidade $\prec_P$ 3-Coloração

Qual é o tamanho do grafo G construído a partir de ϕ?

Quantos vértices têm? Quantas arestas têm?

3-Satisfatibilidade $\prec_P$ 3-Coloração

Qual é o tamanho do grafo G construído a partir de ϕ?

Quantos vértices têm? Quantas arestas têm?

Dado ϕ, podemos construir G em tempo polinomial?

3-Satisfatibilidade $\prec_P$ 3-Coloração

Qual é o tamanho do grafo G construído a partir de ϕ?

Quantos vértices têm? Quantas arestas têm?

Dado ϕ, podemos construir G em tempo polinomial?

É verdade que ϕ é satisfatível $\Leftrightarrow$ G é 3-colorível?

Subset Sum

Problema: Dado um inteiro t e um conjunto S de inteiros, existe um subconjunto de S cuja soma é exatamente t?

Subset Sum

Problema: Dado um inteiro t e um conjunto S de inteiros, existe um subconjunto de S cuja soma é exatamente t?

Exemplo: Se $t = 138457$ e

$S = \{1, 2, 7, 14, 49, 98, 343, 686, 2409, 2793, 16808, 17206, 117705, 117993\}$,

Subset Sum

Problema: Dado um inteiro t e um conjunto S de inteiros, existe um subconjunto de S cuja soma é exatamente t?

Exemplo: Se $t = 138457$ e

$S = \{1, 2, 7, 14, 49, 98, 343, 686, 2409, 2793, 16808, 17206, 117705, 117993\}$,

então o conjunto

$$S' = \{1, 2, 7, 98, 343, 686, 2409, 17206, 117705\}$$

tem soma exatamente t.

Subset Sum

Problema: Dado um inteiro t e um conjunto S de inteiros, existe um subconjunto de S cuja soma é exatamente t?

Exemplo: Se $t = 138457$ e

$S = \{1, 2, 7, 14, 49, 98, 343, 686, 2409, 2793, 16808, 17206, 117705, 117993\}$,

então o conjunto

$$S' = \{1, 2, 7,\ 98, 343, 686, 2409,\ 17206, 117705\}$$

tem soma exatamente t.

Subset Sum

Problema: Dado um inteiro t e um conjunto S de inteiros, existe um subconjunto de S cuja soma é exatamente t?

Exemplo: Se $t = 138457$ e

$S = \{1, 2, 7, 14, 49, 98, 343, 686, 2409, 2793, 16808, 17206, 117705, 117993\}$,

então o conjunto

$$S' = \{1, 2, 7,\ 98, 343, 686, 2409,\ 17206, 117705\}$$

tem soma exatamente t.

Esse problema é um parente próximo do problema da mochila.

Subset Sum e Mochila

Subset Sum: Dado um inteiro t e um conjunto S de inteiros, existe um subconjunto de S cuja soma é exatamente t?

Mochila: Dados inteiros W, V e valores $\{v_1, \ldots, v_n\}$ e pesos $\{w_1, \ldots, w_n\}$, existe um subconjunto I de $[n]$ cuja soma $\sum_{i \in I} w_i \leq W$ e a soma $\sum_{i \in I} v_i \geq V$?

Subset Sum e Mochila

Subset Sum: Dado um inteiro t e um conjunto S de inteiros, existe um subconjunto de S cuja soma é exatamente t?

Mochila: Dados inteiros W, V e valores $\{v_1, \ldots, v_n\}$ e pesos $\{w_1, \ldots, w_n\}$, existe um subconjunto I de $[n]$ cuja soma $\sum_{i \in I} w_i \leq W$ e a soma $\sum_{i \in I} v_i \geq V$?

Consegue mostrar que Mochila está em NP?

Subset Sum e Mochila

Subset Sum: Dado um inteiro t e um conjunto S de inteiros, existe um subconjunto de S cuja soma é exatamente t?

Mochila: Dados inteiros W, V e valores $\{v_1, \ldots, v_n\}$ e pesos $\{w_1, \ldots, w_n\}$, existe um subconjunto I de $[n]$ cuja soma $\sum_{i \in I} w_i \leq W$ e a soma $\sum_{i \in I} v_i \geq V$?

Consegue mostrar que Mochila está em NP?

Consegue mostrar que Subset Sum $\prec_P$ Mochila?

Subset Sum e Mochila

Subset Sum: Dado um inteiro t e um conjunto S de inteiros, existe um subconjunto de S cuja soma é exatamente t?

Mochila: Dados inteiros W, V e valores $\{v_1, \ldots, v_n\}$ e pesos $\{w_1, \ldots, w_n\}$, existe um subconjunto I de $[n]$ cuja soma $\sum_{i \in I} w_i \leq W$ e a soma $\sum_{i \in I} v_i \geq V$?

Consegue mostrar que Mochila está em NP?

Consegue mostrar que Subset Sum $\prec_P$ Mochila?

Fácil:
Para uma instância t e S do Subset Sum, onde $S = \{x_1, \ldots, x_n\}$, tome $W = V = t$ e $w_i = v_i = x_i$ para $i = 1, \ldots, n$.

Subset Sum é NP-completo

Subset Sum: Dado um inteiro t e um conjunto S de inteiros, existe um subconjunto de S cuja soma é exatamente t?

Subset Sum está em NP?

Subset Sum é NP-completo

Subset Sum: Dado um inteiro t e um conjunto S de inteiros, existe um subconjunto de S cuja soma é exatamente t?

Subset Sum está em NP?

Vamos mostrar que 3-Satisfatibilidade $\prec_P$ Subset Sum.

Subset Sum é NP-completo

Subset Sum: Dado um inteiro t e um conjunto S de inteiros, existe um subconjunto de S cuja soma é exatamente t?

Subset Sum está em NP?

Vamos mostrar que 3-Satisfatibilidade $\prec_P$ Subset Sum.

Ou seja, descreveremos um algoritmo polinomial que recebe uma fórmula booleana ϕ com três literais por cláusula, e devolve um inteiro t e um conjunto S de inteiros tais que

$$\phi \text{ é satisfatível} \Leftrightarrow \text{existe } S' \subseteq S \text{ cuja soma é } t.$$

3-Satisfatibilidade

Problema: Dada uma fórmula booleana ϕ nas variáveis $x_1, \ldots, x_n$ em que cada cláusula C_j, para $j = 1, \ldots, m$, tem três literais, existe uma atribuição

$$t : \{x_1, \ldots, x_n\} \rightarrow \{\text{VERDADE}, \text{FALSO}\}$$

que torna ϕ verdadeira?

Podemos assumir que nenhum C_j contém x_i e $\bar{x}_i$.

3-Satisfatibilidade

Problema: Dada uma fórmula booleana ϕ nas variáveis $x_1, \ldots, x_n$ em que cada cláusula C_j, para $j = 1, \ldots, m$, tem três literais, existe uma atribuição

$$t : \{x_1, \ldots, x_n\} \rightarrow \{\text{VERDADE}, \text{FALSO}\}$$

que torna ϕ verdadeira?

Podemos assumir que nenhum C_j contém x_i e $\bar{x}_i$. Por que?

3-Satisfatibilidade

Problema: Dada uma fórmula booleana ϕ nas variáveis $x_1, \ldots, x_n$ em que cada cláusula C_j, para $j = 1, \ldots, m$, tem três literais, existe uma atribuição

$$t : \{x_1, \ldots, x_n\} \to \{\text{VERDADE}, \text{FALSO}\}$$

que torna ϕ verdadeira?

Podemos assumir que nenhum C_j contém x_i e $\bar{x}_i$. Por que?

E que cada variável aparece em alguma cláusula.

3-Satisfatibilidade

Problema: Dada uma fórmula booleana ϕ nas variáveis $x_1, \ldots, x_n$ em que cada cláusula C_j, para $j = 1, \ldots, m$, tem três literais, existe uma atribuição

$$t : \{x_1, \ldots, x_n\} \to \{\text{VERDADE}, \text{FALSO}\}$$

que torna ϕ verdadeira?

Podemos assumir que nenhum C_j contém x_i e $\bar{x}_i$. Por que?

E que cada variável aparece em alguma cláusula. Por que?

3-Satisfatibilidade $\prec_P$ Subset Sum

Problema: Dada uma fórmula booleana ϕ nas variáveis $x_1, \ldots, x_n$ em que cada cláusula C_j, para $j = 1, \ldots, m$, tem três literais, existe uma atribuição

$$t : \{x_1, \ldots, x_n\} \to \{\text{VERDADE}, \text{FALSO}\}$$

que torna ϕ verdadeira?

3-Satisfatibilidade $\prec_P$ Subset Sum

Problema: Dada uma fórmula booleana ϕ nas variáveis $x_1, \ldots, x_n$ em que cada cláusula C_j, para $j = 1, \ldots, m$, tem três literais, existe uma atribuição

$$t : \{x_1, \ldots, x_n\} \to \{\text{VERDADE}, \text{FALSO}\}$$

que torna ϕ verdadeira?

Vamos incluir em S
  dois números para cada variável x_i de ϕ,
  e dois números para cada cláusula C_j de ϕ.

3-Satisfatibilidade $\prec_P$ Subset Sum

Problema: Dada uma fórmula booleana ϕ nas variáveis $x_1, \ldots, x_n$ em que cada cláusula C_j, para $j = 1, \ldots, m$, tem três literais, existe uma atribuição

$$t : \{x_1, \ldots, x_n\} \rightarrow \{\text{VERDADE}, \text{FALSO}\}$$

que torna ϕ verdadeira?

Vamos incluir em S
dois números para cada variável x_i de ϕ,
e dois números para cada cláusula C_j de ϕ.

Portanto total de números em S será $2n + 2m$.

3-Satisfatibilidade $\prec_P$ Subset Sum

Problema: Dada uma fórmula booleana ϕ nas variáveis $x_1, \ldots, x_n$ em que cada cláusula C_j, para $j = 1, \ldots, m$, tem três literais, existe uma atribuição

$$t : \{x_1, \ldots, x_n\} \to \{\text{VERDADE}, \text{FALSO}\}$$

que torna ϕ verdadeira?

Vamos incluir em S
dois números para cada variável x_i de ϕ,
e dois números para cada cláusula C_j de ϕ.

Cada número terá $n + m$ dígitos decimais
sendo que cada dígito corresponde a uma variável x_i,
ou a uma cláusula C_j.

Números em S

Para $C_1 = (x_1 \vee \bar{x}_2 \vee \bar{x}_3)$,
$C_2 = (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$,
$C_3 = (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$,
$C_4 = (x_1 \vee x_2 \vee x_3)$:

		x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1	$=$	1	0	0	1	0	0	1
v_1'	$=$	1	0	0	0	1	1	0
v_2	$=$	0	1	0	0	0	0	1
v_2'	$=$	0	1	0	1	1	1	0
v_3	$=$	0	0	1	0	0	1	1
v_3'	$=$	0	0	1	1	1	0	0

Números em S

Para $C_1 = (x_1 \vee \bar{x}_2 \vee \bar{x}_3)$,
$C_2 = (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$,
$C_3 = (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$,
$C_4 = (x_1 \vee x_2 \vee x_3)$:

		x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1	=	1	0	0	1	0	0	1
v_1'	=	1	0	0	0	1	1	0
v_2	=	0	1	0	0	0	0	1
v_2'	=	0	1	0	1	1	1	0
v_3	=	0	0	1	0	0	1	1
v_3'	=	0	0	1	1	1	0	0

v_i refere-se às ocorrências de x_i nas cláusulas e
v_i' refere-se às ocorrências de $\bar{x}_i$.

Números em S

		x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1	$=$	1	0	0	1	0	0	1
v_1'	$=$	1	0	0	0	1	1	0
v_2	$=$	0	1	0	0	0	0	1
v_2'	$=$	0	1	0	1	1	1	0
v_3	$=$	0	0	1	0	0	1	1
v_3'	$=$	0	0	1	1	1	0	0
s_1	$=$	0	0	0	1	0	0	0
s_1'	$=$	0	0	0	2	0	0	0
s_2	$=$	0	0	0	0	1	0	0
s_2'	$=$	0	0	0	0	2	0	0
s_3	$=$	0	0	0	0	0	1	0
s_3'	$=$	0	0	0	0	0	2	0
s_4	$=$	0	0	0	0	0	0	1
s_4'	$=$	0	0	0	0	0	0	2

Números em S

		x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1	$=$	1	0	0	1	0	0	1
v'_1	$=$	1	0	0	0	1	1	0
v_2	$=$	0	1	0	0	0	0	1
v'_2	$=$	0	1	0	1	1	1	0
v_3	$=$	0	0	1	0	0	1	1
v'_3	$=$	0	0	1	1	1	0	0
s_1	$=$	0	0	0	1	0	0	0
s'_1	$=$	0	0	0	2	0	0	0
s_2	$=$	0	0	0	0	1	0	0
s'_2	$=$	0	0	0	0	2	0	0
s_3	$=$	0	0	0	0	0	1	0
s'_3	$=$	0	0	0	0	0	2	0
s_4	$=$	0	0	0	0	0	0	1
s'_4	$=$	0	0	0	0	0	0	2

Cada par s_j e s'_j está associado à cláusula C_j.

Qual valor de t tomamos?

		x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1	$=$	1	0	0	1	0	0	1
v_1'	$=$	1	0	0	0	1	1	0
v_2	$=$	0	1	0	0	0	0	1
v_2'	$=$	0	1	0	1	1	1	0
v_3	$=$	0	0	1	0	0	1	1
v_3'	$=$	0	0	1	1	1	0	0
s_1	$=$	0	0	0	1	0	0	0
s_1'	$=$	0	0	0	2	0	0	0
s_2	$=$	0	0	0	0	1	0	0
s_2'	$=$	0	0	0	0	2	0	0
s_3	$=$	0	0	0	0	0	1	0
s_3'	$=$	0	0	0	0	0	2	0
s_4	$=$	0	0	0	0	0	0	1
s_4'	$=$	0	0	0	0	0	0	2

Qual valor de t tomamos?

		x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1	$=$	1	0	0	1	0	0	1
v_1'	$=$	1	0	0	0	1	1	0
v_2	$=$	0	1	0	0	0	0	1
v_2'	$=$	0	1	0	1	1	1	0
v_3	$=$	0	0	1	0	0	1	1
v_3'	$=$	0	0	1	1	1	0	0
s_1	$=$	0	0	0	1	0	0	0
s_1'	$=$	0	0	0	2	0	0	0
s_2	$=$	0	0	0	0	1	0	0
s_2'	$=$	0	0	0	0	2	0	0
s_3	$=$	0	0	0	0	0	1	0
s_3'	$=$	0	0	0	0	0	2	0
s_4	$=$	0	0	0	0	0	0	1
s_4'	$=$	0	0	0	0	0	0	2

Para indicar se x_i é verdadeiro ou falso, queremos escolher ou v_i ou v_i' para cada i.

Qual valor de t tomamos?

		x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1	=	1	0	0	1	0	0	1
v_1'	=	1	0	0	0	1	1	0
v_2	=	0	1	0	0	0	0	1
v_2'	=	0	1	0	1	1	1	0
v_3	=	0	0	1	0	0	1	1
v_3'	=	0	0	1	1	1	0	0
s_1	=	0	0	0	1	0	0	0
s_1'	=	0	0	0	2	0	0	0
s_2	=	0	0	0	0	1	0	0
s_2'	=	0	0	0	0	2	0	0
s_3	=	0	0	0	0	0	1	0
s_3'	=	0	0	0	0	0	2	0
s_4	=	0	0	0	0	0	0	1
s_4'	=	0	0	0	0	0	0	2

Para indicar se x_i é verdadeiro ou falso, queremos escolher ou v_i ou v_i' para cada i. Quais os n primeiros dígitos de t?

Qual valor de t tomamos?

		x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1	$=$	1	0	0	1	0	0	1
v_1'	$=$	1	0	0	0	1	1	0
v_2	$=$	0	1	0	0	0	0	1
v_2'	$=$	0	1	0	1	1	1	0
v_3	$=$	0	0	1	0	0	1	1
v_3'	$=$	0	0	1	1	1	0	0
s_1	$=$	0	0	0	1	0	0	0
s_1'	$=$	0	0	0	2	0	0	0
s_2	$=$	0	0	0	0	1	0	0
s_2'	$=$	0	0	0	0	2	0	0
s_3	$=$	0	0	0	0	0	1	0
s_3'	$=$	0	0	0	0	0	2	0
s_4	$=$	0	0	0	0	0	0	1
s_4'	$=$	0	0	0	0	0	0	2

Cada cláusula deve ter pelo menos um literal verdadeiro:

Qual valor de t tomamos?

		x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1	$=$	1	0	0	1	0	0	1
v_1'	$=$	1	0	0	0	1	1	0
v_2	$=$	0	1	0	0	0	0	1
v_2'	$=$	0	1	0	1	1	1	0
v_3	$=$	0	0	1	0	0	1	1
v_3'	$=$	0	0	1	1	1	0	0
s_1	$=$	0	0	0	1	0	0	0
s_1'	$=$	0	0	0	2	0	0	0
s_2	$=$	0	0	0	0	1	0	0
s_2'	$=$	0	0	0	0	2	0	0
s_3	$=$	0	0	0	0	0	1	0
s_3'	$=$	0	0	0	0	0	2	0
s_4	$=$	0	0	0	0	0	0	1
s_4'	$=$	0	0	0	0	0	0	2

Cada cláusula deve ter pelo menos um literal verdadeiro:
um, dois ou três literais verdadeiros por cláusula.

Qual valor de t tomamos?

		x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1	$=$	1	0	0	1	0	0	1
v_1'	$=$	1	0	0	0	1	1	0
v_2	$=$	0	1	0	0	0	0	1
v_2'	$=$	0	1	0	1	1	1	0
v_3	$=$	0	0	1	0	0	1	1
v_3'	$=$	0	0	1	1	1	0	0
s_1	$=$	0	0	0	1	0	0	0
s_1'	$=$	0	0	0	2	0	0	0
s_2	$=$	0	0	0	0	1	0	0
s_2'	$=$	0	0	0	0	2	0	0
s_3	$=$	0	0	0	0	0	1	0
s_3'	$=$	0	0	0	0	0	2	0
s_4	$=$	0	0	0	0	0	0	1
s_4'	$=$	0	0	0	0	0	0	2

Cada cláusula deve ter pelo menos um literal verdadeiro:
um, dois ou três literais verdadeiros por cláusula.
Quais os últimos m dígitos de t?

Números em S e valor de t

		x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1	$=$	1	0	0	1	0	0	1
v_1'	$=$	1	0	0	0	1	1	0
v_2	$=$	0	1	0	0	0	0	1
v_2'	$=$	0	1	0	1	1	1	0
v_3	$=$	0	0	1	0	0	1	1
v_3'	$=$	0	0	1	1	1	0	0
s_1	$=$	0	0	0	1	0	0	0
s_1'	$=$	0	0	0	2	0	0	0
s_2	$=$	0	0	0	0	1	0	0
s_2'	$=$	0	0	0	0	2	0	0
s_3	$=$	0	0	0	0	0	1	0
s_3'	$=$	0	0	0	0	0	2	0
s_4	$=$	0	0	0	0	0	0	1
s_4'	$=$	0	0	0	0	0	0	2
t	$=$	1	1	1	4	4	4	4

Subset Sum é NP-completo

3-Satisfatibilidade $\prec_P$ Subset Sum.

A construção destes números pode ser feita em tempo $\mathrm{O}(nm)$.
Portanto é polinomial no tamanho de ϕ.

Assim descrevemos um algoritmo polinomial que
recebe uma fórmula booleana ϕ com três literais por cláusula,
e devolve um inteiro t e um conjunto S de inteiros tais que

$$\phi \text{ é satisfatível} \Leftrightarrow \text{existe } S' \subseteq S \text{ cuja soma é } t.$$

Subset Sum é NP-completo

3-Satisfatibilidade $\prec_P$ Subset Sum.

A construção destes números pode ser feita em tempo $\mathrm{O}(nm)$.
Portanto é polinomial no tamanho de ϕ.

Assim descrevemos um algoritmo polinomial que
recebe uma fórmula booleana ϕ com três literais por cláusula,
e devolve um inteiro t e um conjunto S de inteiros tais que

$$\phi \text{ é satisfatível} \Leftrightarrow \text{existe } S' \subseteq S \text{ cuja soma é } t.$$

Portanto Subset Sum é NP-completo.