Politechnika Wrocławska

Struktury danych i złożoność obliczeniowa
Wykład 7

Prof. dr hab. inż. Jan Magott

Problem plecakowy

Wersja decyzyjna - problem NP-zupełny

Wersja optymalizacyjna - problem NP-trudny

Liczba możliwych rozwiązań jest (2^n)

Propozycja algorytmu przybliżonego o wielomianowej złożoności

Problem plecakowy - wersja decyzyjna

Dane:

Skończony zbiór elementów $A = \{a_1, a_2, \dots, a_n\}$.

Rozmiar $s(a_i) > 0$ i waga (wartość) $w(a_i) > 0$ elementu a_i .

Pojemność plecaka $b > 0$ i stała $y > 0$.

Zadanie:

Czy istnieje podzbiór $A' \subset A$ taki, że:

$$\sum_{a_i \in A'} s(a_i) \leq b$$

$$\sum_{a_i \in A'} w(a_i) \geq y \quad ?$$

Algorytm przybliżony dla wersji optymalizacyjnej

1. Dla każdego elementu $i \in \{\overline{1,n}\}$ oblicz $p_i = w(a_i)/s(a_i)$.
   $A' = \emptyset$ i $S = 0$.
2. Wyznacz sekwencję $\sigma =< \sigma(1), \dots, \sigma(n) >$ taką, że
   $p_{\sigma(j)} \geq p_{\sigma(j+1)}$ dla $j \in \{\overline{1,n-1}\}$.
3. Dla $j \in \{\overline{1,n}\}$ wykonuj: Jeżeli $S + s(a_{\sigma(j)}) \leq b$, to
   $A' := A' \cup \{a_{\sigma(j)}\}$ i $S := S + s(a_{\sigma(j)})$.
4. Stop

A' jest przybliżonym rozwiązaniem uzyskanym algorytmem o złożoności wielomianowej.

i	$s(a_i)$	$w(a_i)$
1	1	9
2	2	8
3	6	7
4	7	10
5	1	8
6	4	7

b=12

i	$s(a_i)$	$w(a_i)$	$w(a_i)/s(a_i)$
1	1	9	9
2	2	8	4
3	6	7	1,17
4	7	10	1,43
5	1	8	8
6	4	7	1,75

b=12

$$\boldsymbol{S = < 1, 5, 2, 6, 4, 3 >}$$

$$\boldsymbol{S} =< \mathbf{1}, \mathbf{5}, \mathbf{2}, \mathbf{6}, \mathbf{4}, \mathbf{3} >$$

Weź a_1 i sprawdź czy $S + s(a_1) \leq b$ czyli $0 + 1 \leq 12.$

Zatem $A' = \{a_1\}$ i $S = 1.$

Weź a_5 i sprawdź czy $S + s(a_5) \leq b$ czyli $1 + 1 \leq 12.$

Zatem $A' = \{a_1, a_5\}$ i $S = 2.$

Weź a_2 i sprawdź czy $S + s(a_2) \leq b$ czyli $2 + 2 \leq 12.$

Zatem $A' = \{a_1, a_5, a_2\}$ i $S = 4.$

Weź a_6 i sprawdź czy $S + s(a_6) \leq b$ czyli $4 + 4 \leq 12.$

Zatem $A' = \{a_1, a_5, a_2, a_6\}$ i $S = 8.$

Weź a_4 i sprawdź czy $S + s(a_4) \leq b$ czyli $8 + 7 \leq 12.$

Zatem $A' = \{a_1, a_5, a_2, a_6\}$ i $S = 8.$

Weź a_3 i sprawdź czy $S + s(a_3) \leq b$ czyli $8 + 6 \leq 12.$

Zatem $A' = \{a_1, a_5, a_2, a_6\}$ i $S = 8.$

Zatem $A' = \{a_1, a_5, a_2, a_6\}$ i $S = 8$.

Rozwiązanie:

$$A' = \{a_1, a_5, a_2, a_6\}$$

zajętość plecaka:

$$S = 8$$

sumaryczna wartość zapakowanych elementów:

$$\sum_{a_i \in A'} w(a_i) = 9 + 8 + 8 + 7 = 32$$

Jaka jest złożoność obliczeniowa algorytmu?

1. Dla każdego elementu $i \in \{\overline{1,n}\}$ oblicz $p_i = w(a_i)/s(a_i)$.
   $A' = \emptyset$ i $S = 0$.
2. Wyznacz sekwencję $\sigma =< \sigma(1), \ldots, \sigma(n) >$ taką, że $p_{\sigma(j)} \geq p_{\sigma(j+1)}$ dla $j \in \{\overline{1,n-1}\}$.
3. Dla $j \in \{\overline{1,n}\}$ wykonuj: Jeżeli $S + s(a_{\sigma(j)}) \leq b$, to
   $A' := A' \cup \{a_{\sigma(j)}\}$ i $S := S + s(a_{\sigma(j)})$.
4. Stop

Jaka jest złożoność obliczeniowa algorytmu?

1. Dla każdego elementu $i \in \{\overline{1,n}\}$ oblicz $p_i = w(a_i)/s(a_i)$.
   $A' = \emptyset$ i $S = 0$. $\boldsymbol{O(n)}$
2. Wyznacz sekwencję $\sigma =< \sigma(1), \dots, \sigma(n) >$ taką, że $p_{\sigma(j)} \geq p_{\sigma(j+1)}$ dla $j \in \{\overline{1, n-1}\}$.
3. Dla $j \in \{\overline{1,n}\}$ wykonuj: Jeżeli $S + s(a_{\sigma(j)}) \leq b$, to
   $A' \coloneqq A' \cup \{a_{\sigma(j)}\}$ i $S \coloneqq S + s(a_{\sigma(j)})$.
4. Stop

Jaka jest złożoność obliczeniowa algorytmu?

1. Dla każdego elementu $i \in \{\overline{1,n}\}$ oblicz $p_i = w(a_i)/s(a_i)$.
   $A' = \emptyset$ i $S = 0$. $\boldsymbol{O(n)}$
2. Wyznacz sekwencję $\sigma =< \sigma(1), \dots, \sigma(n) >$ taką, że
   $p_{\sigma(j)} \geq p_{\sigma(j+1)}$ dla $j \in \{\overline{1, n-1}\}$. $\boldsymbol{O(n \log n)}$
3. Dla $j \in \{\overline{1,n}\}$ wykonuj: Jeżeli $S + s(a_{\sigma(j)}) \leq b$, to
   $A' := A' \cup \{a_{\sigma(j)}\}$ i $S := S + s(a_{\sigma(j)})$.
4. Stop

Jaka jest złożoność obliczeniowa algorytmu?

1. Dla każdego elementu $i \in \{\overline{1,n}\}$ oblicz $p_i = w(a_i)/s(a_i)$.
   $A' = \emptyset$ i $S = 0$. $\boldsymbol{O(n)}$
2. Wyznacz sekwencję $\sigma =< \sigma(1), \ldots, \sigma(n) >$ taką, że $p_{\sigma(j)} \geq p_{\sigma(j+1)}$ dla $j \in \{\overline{1, n-1}\}$. $\boldsymbol{O(n \log n)}$
3. Dla $j \in \{\overline{1,n}\}$ wykonuj: Jeżeli $S + s(a_{\sigma(j)}) \leq b$, to
   $A' := A' \cup \{a_{\sigma(j)}\}$, $S := S + s(a_{\sigma(j)})$ i $W := W + w(a_{\sigma(j)})$. $\boldsymbol{O(n)}$
4. Stop

Złożoność obliczeniowa algorytmu

$$\boldsymbol{O}(\boldsymbol{n}) + \boldsymbol{O}(\boldsymbol{n}\log\boldsymbol{n}) + \boldsymbol{O}(\boldsymbol{n}) = \boldsymbol{O}(\boldsymbol{n}\log\boldsymbol{n})$$

Czy rozwiązanie:

$$A' = \{a_1, a_5, a_2, a_6\}$$

zajętość plecaka:

$$S = 8$$

sumaryczna wartość zapakowanych elementów:

$$\sum_{a_i \in X} w(a_i) = 9 + 8 + 8 + 7 = 32$$

jest optymalnym ?

i	$s(a_i)$	$w(a_i)$
1	1	9
2	2	8
3	6	7
4	7	10
5	1	8
6	4	7

b=12

Lepszym rozwiązaniem jest:

$$A' = \{a_1, a_5, a_2, a_4\}$$

zajętość plecaka:

$$S = 11$$

sumaryczna wartość zapakowanych elementów:

$$\sum_{a_i \in X} w(a_i) = 9 + 8 + 8 + 10 = 35$$

Inne rozwiązanie problemu plecakowego

$c(j,S)$ największa wartość upakowania plecaka o pojemności S elementami ze zbioru $\{a_1, \dots, a_j\}$.

$c(n,b)$ jest rozwiązaniem problemu plecakowego.

Własności

$$c(0,S) = 0$$
$$c(j,0) = 0$$
$$S1 \le S2 \;\Rightarrow\; c(j,S1) \le c(j,S2)$$
$$j1 \le j2 \;\Rightarrow\; c(j1,S) \le c(j2,S)$$

Inne rozwiązanie problemu plecakowego

$c(j,S)$ największa wartość upakowania plecaka o pojemności S elementami ze zbioru $\{a_1,\dots,a_j\}$.

$$c(j,S)=\begin{cases} 0 & if\ \ j=0\ lub\ S=0 \\ c(j-1,S) & if\quad s(a_j)>S \\ \max\left\{c(j-1,S),c\left(j-1,S-s(a_j)\right)+w(a_j)\right\} & if\ s(a_j)\le S \end{cases}$$

Jeśli $s(a_j)>S$, to elementu a_j nie można umieścić w plecaku o pojemności S.

Jeśli $s(a_j)\le S$, to element a_j zmieści się w plecaku.

Jeśli $c(j-1,S)>c\left(j-1,S-s(a_j)\right)+w(a_j)$, to maksymalna wartość plecaka o pojemności S bez elementu a_j jest większa niż z nim.

Algorytm wyznaczania optymalnej wartości plecaka

Dla $j = 0, \dots, n$ wykonaj

Dla $S = 0, \dots, b$

wyznacz $c(j, S)$

$c(n, b)$ jest optymalną wartością plecaka

j/S	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0												
2	0												
3	0												
4	0												
5	0												
6	0												

b=12

i	$s(a_i)$	$w(a_i)$
1	1	9
2	2	8
3	6	7
4	7	10
5	1	8
6	4	7

b=12

Politechnika

i	$s(a_i)$	$w(a_i)$
1	1	9
2	2	8
3	6	7
4	7	10
5	1	8
6	4	7

j/S	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	9	9	9	9	9	9	9	9	9	9	9	9
2	0	9	9	17	17	17	17	17	17	17	17	17	17
3	0	9	9	17	17	17	17	17	17	24	24	24	24
4	0	9	9	17	17	17	17	17	19	24	27	27	27
5	0	9	17	17	25	25	25	25	25	27	32	35	35
6	0	9	17	17	25	25	25	25	32	32	32	35	35

b=12

Optymalnym rozwiązaniem jest:

$$A' = \{a_1, a_5, a_2, a_4\}$$

zajętość plecaka:

$$S = 11$$

sumaryczna wartość zapakowanych elementów:

$$\sum_{a_i \in X} w(a_i) = 9 + 8 + 8 + 10 = 35$$

Cel

Z rozpatrywanej tabeli należy wyznaczyć rozwiązanie.

i	$s(a_i)$	$w(a_i)$
1	1	9
2	2	8
3	6	7
4	7	10
5	1	8
6	4	7

b=12

$c(n, b)$ jest optymalną wartością plecaka czyli $c(6{,}12) = 35$

$s(a_6) = 4 \leq S = b = 12$

$c(6{,}12) = c(5{,}12)$

czyli

$$\boldsymbol{c(j, S)} = \begin{cases} 0 & if \ \ j = 0 \ lub \ S = 0 \\ \mathrm{c(j-1, S)} & if \quad s(a_j) > S \\ \max\{\boldsymbol{c(j-1, S)}, c(j-1, S - s(a_j)) + w(a_j)\} & if \ \boldsymbol{s(a_j) \leq S} \end{cases}$$

Zatem $a_6 \notin A'$.

$c(5{,}12) = 35$

$s(a_5) = 1 \leq S = b = 12$

$c(5{,}12) = c(4{,}11) + w(a_5) \ = 27 + 8$

czyli

$$\boldsymbol{c(j, S)} = \begin{cases} 0 & if \ \ j = 0 \ lub \ S = 0 \\ c(j-1, S) & if \quad s(a_j) > S \\ \max\left\{c(j-1, S), \boldsymbol{c}\left(\boldsymbol{j-1, S - s(a_j)}\right) + \boldsymbol{w(a_j)}\right\} & if \ \boldsymbol{s(a_j) \leq S} \end{cases}$$

Zatem $a_5 \in A'$.

j/S	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	9	9	9	9	9	9	9	9	9	9	9	9
2	0	9	9	17	17	17	17	17	17	17	17	17	17
3	0	9	9	17	17	17	17	17	17	24	24	24	24
4	0	9	9	17	17	17	17	17	19	24	27	27	27
5	0	9	17	25	25	25	25	25	25	27	32	35	35
6	0	9	17	25	25	25	25	25	32	32	32	35	35

b=12

Programowanie dynamiczne (backtracking)

Jaki problem NP-zupełny możemy rozwiązać w podobny sposób ?

Jaki problem NP-zupełny możemy rozwiązać w podobny sposób ?

PROBLEM PODZIAŁU ZBIORU

Algorytm wyznaczania optymalnej wartości plecaka wraz z rozwiązaniem

Dla $j = 0, \dots, n$ wykonaj $n + 1$ wykonań

Dla $S = 0, \dots, b$ $b + 1$ wykonań

oblicz $c(j, S)$ $O(1)$

Wyznacz rozwiązanie metodą „backtracking” $O(n)$

Złożoność obliczeniowa algorytmu

$$(n+1)(b+1)O(1) + O(n) = O(nb) + O(n) = O(nb)$$

Czy przedstawiony algorytm dla problemu NP-zupełnego jest wielomianowym?

Jego złożoność $O(nb)$ zależy od wartości pojemności plecaka czyli parametru liczbowego, który nie jest rozmiarem problemu.

Niektóre problemy NP-zupełne można dla spotykanych w praktyce danych rozwiązać w stosunkowo niewielkim czasie za pomocą tzw. algorytmów pseudowielomianowych.

Złożoność obliczeniowa algorytmów pseudowielomianowych jest ograniczona od góry przez wielomian p zależny od rozmiaru instancji $N(I)$ oraz od maksymalnej wartości $Max(I)$ liczb występujących w tej instancji $O\left(p\left(N(I), Max(I)\right)\right)$.

Wniosek

Algorytm wielomianowy jest algorytmem pseudowielomianowym.

Liczba wszystkich rozwiązań w problemie plecakowym jest 2^n.

Złożoność analizowanego algorytmu $O(nb)$.

Istnieje instancja o własności $2^n < b$.

Zatem powyższy algorytm zastosowany do tej instancji ma złożoność wykładniczą w funkcji rozmiarów problemu.

Jednak często $2^n \gg b$.

Problem π jest ***problemem liczbowym***, jeśli nie istnieje wielomian p taki, że $Max(I) \leq p\big(N(I)\big)$ dla wszystkich instancji $I \in D_{\pi}$.

Gdyby powyższy wymóg był spełniony, to byłoby silne ograniczenie na wartości liczbowe danych.

Problem nie spełniający powyższego wymogu jest nieliczbowym.

Przykłady problemów liczbowych: Problem podziału, Problem plecakowy, Problem komiwojażera.

Przykłady problemów nieliczbowych: problem spełnialności wyrażeń logicznych, Problem cyklu Hamiltona, problem najdłuższej ścieżki.

Problem liczbowy π to taki problem, dla którego nie istnieje wielomian p taki, że $Max(I) \leq p(N(I))$ dla każdej instancji $I \in D_\pi$.

Twierdzenie

Jeżeli problem NP-zupełny jest nieliczbowym, to nie istnieje pseudowielomianowy algorytm jego rozwiązania.

Algorytmy pseudowielomianowe można ewentualnie zbudować tylko dla problemów liczbowych.

Próba rozróżnienia wśród NP-zupełnych problemów decyzyjnych tych, dla których można zbudować algorytmy pseudowielomianowe od tych, dla których takich algorytmów nie można skonstruować wymaga definicji problemu silnie NP-zupełnego.

Dla dowolnego problemu decyzyjnego π i dowolnego wielomianu p, określonego dla liczb całkowitych, niech π_p oznacza ***podproblem problemu** π **otrzymany przez ograniczenie*** dziedziny D_π tylko ***do tych instancji, dla których** $\boldsymbol{Max(I) \leq p(N(I))}$*.

π_p nie jest problemem liczbowym.

Problem decyzyjny** $\boldsymbol{\pi}$ **jest silnie NP-zupełny, jeśli należy do klasy NP i istnieje wielomian p określony dla liczb całkowitych, dla którego π_p jest NP-zupełny.

Wniosek 1. Jeśli problem π jest NP-zupełny i nie jest problemem liczbowym, to jest silnie NP-zupełny.

Wniosek 2. Jeśli problem jest silnie NP-zupełny, to nie istnieje dla niego pseudowielomianowy algorytm rozwiązania, jeśli $P \neq NP$.

Problem podziału, Problem plecakowy nie są silnie NP-zupełnymi.

Problem trójpodziału zbioru

Dane:

- $C = \{c_1, \dots, c_i, \dots, c_{3k}\}$ - zbiór $3k$ elementów,
- Rozmiar $s(c_i) > 0$ elementu c_i, gdzie $s(c_i) \in N_+$, $N_+ = \{1,2,\dots\}$,

$$B \in N_+,\ \frac{B}{4} < s(c_i) < \frac{\mathrm{B}}{2},\ \sum_{i=1}^{3k} s(c_i) = kB.$$

Pytanie:

Czy istnieje podział zbioru C na k podzbiorów takich, że

$i \neq j => S_i \cap S_j = \emptyset$

$$1 \leq i \leq k \quad \sum_{c_j \epsilon C_i} s(c_j) = B\ ?$$

Problem trójpodziału jest silnie NP-zupełny.

Dowód silnej NP-zupełności jest trudny.

Dla uproszczenia dowodu wprowadzono pojęcie transformacji pseudowielomianowej.

Transformacją pseudowielomianową problemu decyzyjnego $\boldsymbol{\pi_2}$ do problemu decyzyjnego $\boldsymbol{\pi_1}$ jest funkcja $f\colon D_{\pi_2} \rightarrow D_{\pi_1}$ spełniająca warunki:

1. Dla każdej instancji $I_2 \epsilon D_{\pi_2}$ odpowiedź brzmi „tak”, wtedy i tylko wtedy, gdy dla instancji $f(I_2)$ odpowiedź również jest „tak”,

2. Czas obliczenia funkcji f przez DMT dla każdej instancji $I_2 \epsilon D_{\pi_2}$ jest ograniczony od góry przez wielomian p od $N_2(I_2)$ i $Max_2(I_2)$ czyli jest $O\big(p(N_2(I_2), Max_2(I_2))\big)$,

3. Istnieje wielomian q_1 taki, że dla każdej $I_2 \epsilon D_{\pi_2}$ jest

$q_1(N_1(f(I_2)) \geq N_2(I_2)$,

4. Istnieje wielomian q_2 taki, że dla każdej $I_2 \epsilon D_{\pi_2}$ jest

$Max_1(f(I_2)) \leq q_2(Max_2(I_2), N_2(I_2))$.

(3) Rozmiar instancji po transformacji nie może zmaleć wykładniczo.

(4) Największa wartość liczbowa danych instancji po transformacji nie może wzrosnąć wykładniczo.

Dowód silnej NP-zupełności problemu π:

1. Pokazać $\pi \in NP$,
2. Wybrać inny NP-zupełny problem π',
3. Wykonać pseudowielomianową transformację problemu π' do problemu π .

Problem komiwojażera

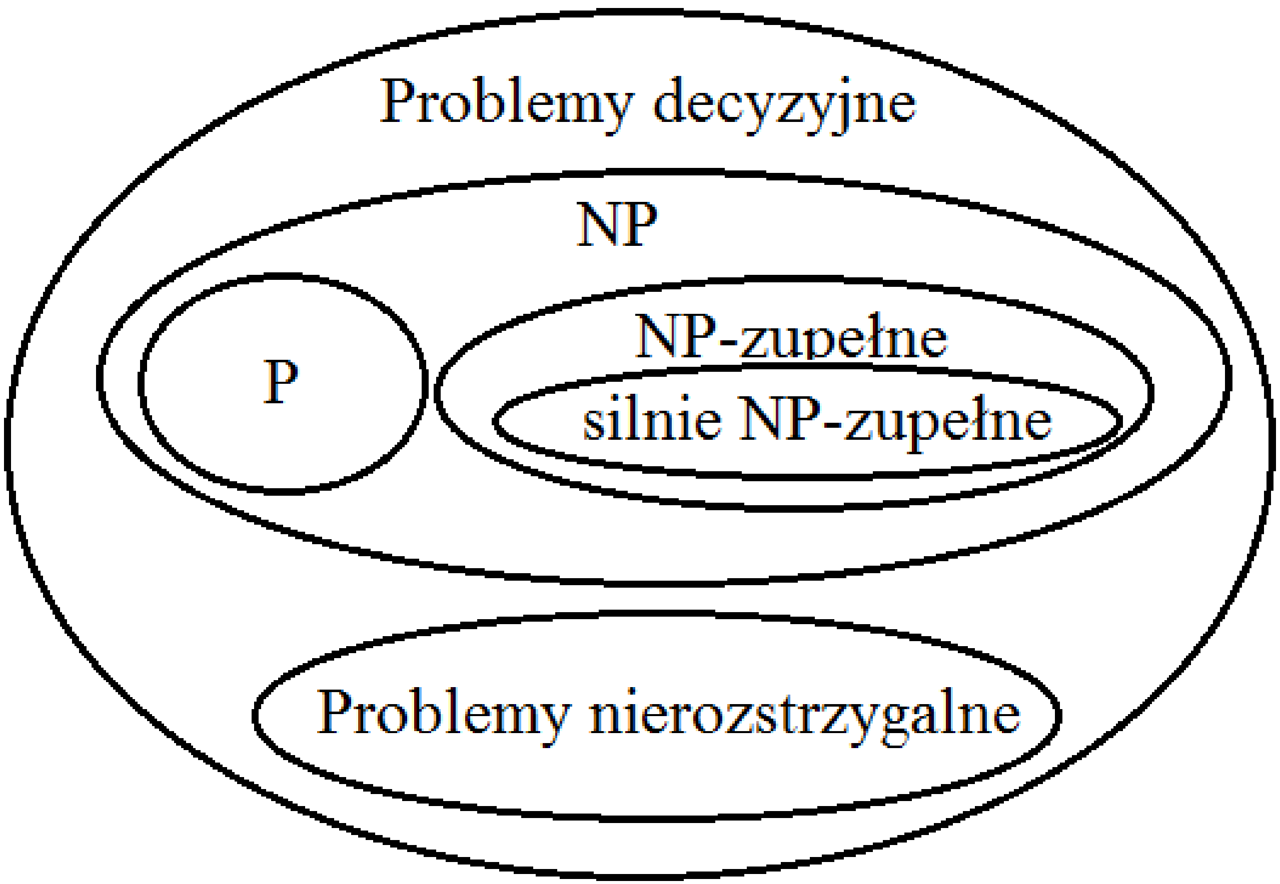
i

Problem trójpodziału

są

liczbowymi problemami silnie NP-zupełnymi.

Założenie: *P≠NP*



Jeśli $P = NP$

to

silnie NP-zupełne = *NP-zupełne* = *NP* = *P*

Granica między *Problemami klasy P*, a klasą *Problemów NP-zupełnych* jest niekiedy zaskakująca.

Problem cyklu Eulera $(\in P)$

Dane:

Graf nieskierowany $G =< V, E >$

Pytanie:

Czy istnieje cykl (zamknięta droga) przechodzący przez każdą krawędź dokładnie jeden raz?

Problem cyklu Hamiltona $(\in NP - zupełne)$

Dane:

Graf nieskierowany $G =< V, E >$

Pytanie:

Czy istnieje cykl przechodzący przez każdy wierzchołek dokładnie jeden raz?

Problem najkrótszej ścieżki w grafie z długościami krawędzi $(\in P)$

Dane:

Graf nieskierowany $G =< V, E, L >$, gdzie $L(e) \in N_+$ jest długością krawędzi $e \in E$, z wyróżnionymi wierzchołkami $v, u \in V$. $B \in N_+$.

Pytanie:

Czy istnieje droga prosta (bez cykli) z v do u o długości nie większej niż B ?

Problem najdłuższej ścieżki w grafie z długościami krawędzi $(\in NP - zupełne)$

Dane:

Graf nieskierowany $G =< V, E, L >$, gdzie $L(e) \in N_+$ jest długością krawędzi $e \in E$, z wyróżnionymi wierzchołkami $v, u \in V$. $B \in N_+$.

Pytanie:

Czy istnieje droga prosta z v do u o długości nie mniejszej niż B ?

Problem trójkolorowania krawędzi grafu $(\in P)$

Dane:

Graf nieskierowany $G = < V, E >$ i trzy kolory.

Pytanie:

Czy istnieje taki sposób pokolorowania krawędzi, że każde dwie incydentne krawędzie mają różne kolory?

Problem trójkolorowania wierzchołków grafu $(\in NP - \text{zupełne})$

Dane:

Graf nieskierowany $G = < V, E >$ i trzy kolory.

Pytanie:

Czy istnieje taki sposób pokolorowania wierzchołków, że każde dwa incydentne wierzchołki (połączone krawędzią) mają różne kolory?

Problem przeszukiwania (zbiór instancji i zbiór rozwiązań dla każdej instancji) $\boldsymbol{\pi}$ ***jest silnie NP-trudny***, jeśli należy do klasy NP a jego podproblem spełniający dla pewnego wielomianu p ograniczenie $\boldsymbol{Max(I)} \leq \boldsymbol{p(N(I))}$ jest NP-trudny.

Problem optymalizacyjny jest silnie NP-trudny, jeśli jego wersja decyzyjna jest silnie NP-zupełna.