Złożoność obliczeniowa algorytmów

Problemy **NP**-zupełne

Kordian A. Smoliński

Wydział Fizyki i Informatyki Stosowanej

2024/2025

Problemy **NP**-zupełne

Treść wykładu

1. SAT
   - 3SAT
   - MAX2SAT
2. Problemy grafowe
   - Pokrycie wierzchołkowe
   - Zbiór niezależny i klika
   - Cykl Hamiltona i droga Hamiltona
3. Problemy na zbiorach i liczbach
   - Skojarzenie trójdzielne
   - Problemy dot. pokryć
   - Problemy liczbowe

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy **NP**-zupełne

Aby dowieść, że język (problem decyzyjny) L jest **NP**-zupełny, należy:

Problemy **NP**-zupełne

Aby dowieść, że język (problem decyzyjny) L jest **NP**-zupełny, należy:

1. dowieść, że język L należy do klasy **NP**;

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy **NP**-zupełne

Aby dowieść, że język (problem decyzyjny) L jest **NP**-zupełny, należy:

1. dowieść, że język L należy do klasy **NP**;
2. skonstruować redukcję wybranego znanego języka **NP**-zupełnego L' do L.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy **NP**-zupełne

Aby dowieść, że język (problem decyzyjny) L jest **NP**-zupełny, należy:

1. dowieść, że język L należy do klasy **NP**;
2. skonstruować redukcję wybranego znanego języka **NP**-zupełnego L' do L.

Rozpoczynamy zatem od dowiedzenia **NP**-zupełności kilku klasycznych problemów.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

SAT

Definicja

Formuła rachunku zdań może być:

SAT

Definicja

Formuła rachunku zdań może być:

- zmienną zdaniową x_i;

SAT

Definicja

Formuła rachunku zdań może być:

- zmienną zdaniową x_i;
- negacją formuły rachunku zdań ϕ_i;

SAT

Definicja

Formuła rachunku zdań może być:

- zmienną zdaniową x_i;
- negacją formuły rachunku zdań ϕ_i;
- alternatywą formuł rachunku zdań $\phi_i \vee \phi_j$;

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

SAT

Definicja

Formuła rachunku zdań może być:

- zmienną zdaniową x_i;
- negacją formuły rachunku zdań ϕ_i;
- alternatywą formuł rachunku zdań $\phi_i \vee \phi_j$;
- koniunkcją formuł rachunku zdań $\phi_i \wedge \phi_j$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

SAT

Definicja

Formuła rachunku zdań może być:

- zmienną zdaniową x_i;
- negacją formuły rachunku zdań ϕ_i;
- alternatywą formuł rachunku zdań $\phi_i \vee \phi_j$;
- koniunkcją formuł rachunku zdań $\phi_i \wedge \phi_j$.

Definicja

Wartościowanie logiczne to odwzorowanie ze skończonego zbioru zmiennych zdaniowych w zbiór wartości logicznych $\{\textbf{prawda}, \textbf{fałsz}\}$.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

SAT

Definicja

Formuła rachunku zdań może być:

- zmienną zdaniową x_i;
- negacją formuły rachunku zdań ϕ_i;
- alternatywą formuł rachunku zdań $\phi_i \vee \phi_j$;
- koniunkcją formuł rachunku zdań $\phi_i \wedge \phi_j$.

Definicja

Wartościowanie logiczne to odwzorowanie ze skończonego zbioru zmiennych zdaniowych w zbiór wartości logicznych {**prawda**, **fałsz**}.

Definicja

Wartościowanie T spełnia formułę ϕ, $T \vDash \phi$, jeżeli dla tego wartościowania $\phi \mapsto$ **prawda**.

SAT

Definicja

ϕ jest spełnialna, jeżeli $\exists T: T \vDash \phi$.

SAT

Definicja

ϕ jest spełnialna, jeżeli $\exists T: \; T \vDash \phi$.

Definicja

ϕ jest prawdziwa, $\vDash \phi$, jeżeli $\forall T: \; T \vDash \phi$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

SAT

Definicja

ϕ jest spełnialna, jeżeli $\exists T: \; T \vDash \phi$.

Definicja

ϕ jest prawdziwa, $\vDash \phi$, jeżeli $\forall T: \; T \vDash \phi$.

Definicje

literał — zmienna zdaniowa x_i lub jej negacja $\overline{x_i}$;

SAT

Definicja

ϕ jest spełnialna, jeżeli $\exists T: \; T \vDash \phi$.

Definicja

ϕ jest prawdziwa, $\vDash \phi$, jeżeli $\forall T: \; T \vDash \phi$.

Definicje

literał — zmienna zdaniowa x_i lub jej negacja $\overline{x_i}$;

klauzula — alternatywa C_i co najmniej jednego literału;

SAT

Definicja

ϕ jest spełnialna, jeżeli $\exists T: \; T \vDash \phi$.

Definicja

ϕ jest prawdziwa, $\vDash \phi$, jeżeli $\forall T: \; T \vDash \phi$.

Definicje

literał — zmienna zdaniowa x_i lub jej negacja $\overline{x_i}$;

klauzula — alternatywa C_i co najmniej jednego literału;

implikant — koniunkcja D_i co najmniej jednego literału;

SAT

Definicja

ϕ jest spełnialna, jeżeli $\exists T: \; T \vDash \phi$.

Definicja

ϕ jest prawdziwa, $\vDash \phi$, jeżeli $\forall T: \; T \vDash \phi$.

Definicje

literał — zmienna zdaniowa x_i lub jej negacja $\overline{x_i}$;

klauzula — alternatywa C_i co najmniej jednego literału;

implikant — koniunkcja D_i co najmniej jednego literału;

koniunktywna postać normalna — koniunkcja skończonej liczby klauzul: $\phi = \bigwedge_{i=1}^{n} C_i$ $(n \geqslant 1)$;

SAT

Definicja

ϕ jest spełnialna, jeżeli $\exists T: \; T \vDash \phi$.

Definicja

ϕ jest prawdziwa, $\vDash \phi$, jeżeli $\forall T: \; T \vDash \phi$.

Definicje

literał — zmienna zdaniowa x_i lub jej negacja $\overline{x_i}$;

klauzula — alternatywa C_i co najmniej jednego literału;

implikant — koniunkcja D_i co najmniej jednego literału;

koniunktywna postać normalna — koniunkcja skończonej liczby klauzul: $\phi = \bigwedge_{i=1}^{n} C_i$ $(n \geqslant 1)$;

dysjunktywna postać normalna — alternatywa skończonej liczby implikantów: $\phi = \bigvee_{i=1}^{n} D_i$ $(n \geqslant 1)$.

Twierdzenie

Dla każdej formuły rachunku zdań istnieje równoważna formuła w koniunktywnej postaci normalnej oraz równoważna formuła w dysjunktywnej postaci normalnej.

SAT

Twierdzenie

Dla każdej formuły rachunku zdań istnieje równoważna formuła w koniunktywnej postaci normalnej oraz równoważna formuła w dysjunktywnej postaci normalnej.

Problem (SAT)

Czy dla danej formuły rachunku zdań zapisanej w koniunktywnej postaci normalnej istnieje wartościowanie ją spełniające?

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

SAT

Twierdzenie

Dla każdej formuły rachunku zdań istnieje równoważna formuła w koniunktywnej postaci normalnej oraz równoważna formuła w dysjunktywnej postaci normalnej.

Problem (SAT)

Czy dla danej formuły rachunku zdań zapisanej w koniunktywnej postaci normalnej istnieje wartościowanie ją spełniające?

Fakt (Cook–Lewin)

Problem SAT *jest* **NP***-zupełny.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

3SAT

Problem (3SAT)

Czy dla danej formuły rachunku zdań zapisanej w koniunktywnej postaci normalnej, w której wszystkie klauzule mają co najwyżej trzy literały, istnieje wartościowanie ją spełniające?

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

3SAT

Problem (3SAT)

Czy dla danej formuły rachunku zdań zapisanej w koniunktywnej postaci normalnej, w której wszystkie klauzule mają co najwyżej trzy literały, istnieje wartościowanie ją spełniające?

Fakt

Problem 3SAT *jest* **NP***-zupełny.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

3SAT

Problem (3SAT)

Czy dla danej formuły rachunku zdań zapisanej w koniunktywnej postaci normalnej, w której wszystkie klauzule mają co najwyżej trzy literały, istnieje wartościowanie ją spełniające?

Fakt

Problem 3SAT *jest* **NP***-zupełny.*

Dowód.

3SAT jest podproblemem SAT, który jest w **NP**, więc sam jest w **NP**.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

3SAT

Problem (3SAT)

Czy dla danej formuły rachunku zdań zapisanej w koniunktywnej postaci normalnej, w której wszystkie klauzule mają co najwyżej trzy literały, istnieje wartościowanie ją spełniające?

Fakt

Problem 3SAT *jest* **NP***-zupełny.*

Dowód.

3SAT jest podproblemem SAT, który jest w **NP**, więc sam jest w **NP**.
Redukcja z SAT do 3SAT: Niech $\phi = \bigwedge_{i=1}^{m} C_i$ nad $x_1, \ldots, x_n$, gdzie $C_i = \bigvee_{j=1}^{k} x_{i_j}$, a x_{i_j} są parami różne ($j = 1, \ldots, k$).

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Dowód (dokończenie).

Jeżeli w C_i mamy $k > 3$, to dodajemy $k - 3$ nowych zmiennych $y_{i_2}, \ldots, y_{i_{k-2}}$ i zastępujemy C_i przez

$$C_i' = (x_{i_1} \vee x_{i_2} \vee y_{i_2}) \wedge (\neg y_{i_2} \vee x_{i_3} \vee y_{i_3}) \wedge \cdots \wedge (\neg y_{i_{k-2}} \vee x_{i_{k-1}} \vee x_{i_k}).$$

Dowód (dokończenie).

Jeżeli w C_i mamy $k > 3$, to dodajemy $k - 3$ nowych zmiennych $y_{i_2}, \ldots, y_{i_{k-2}}$ i zastępujemy C_i przez

$$C_i' = (x_{i_1} \vee x_{i_2} \vee y_{i_2}) \wedge (\neg y_{i_2} \vee x_{i_3} \vee y_{i_3}) \wedge \cdots \wedge (\neg y_{i_{k-2}} \vee x_{i_{k-1}} \vee x_{i_k}).$$

Jeżeli C_i jest spełnialna, to można dobrać wartościowanie dla $y_{i_2}, \ldots, y_{i_{k-2}}$, aby C_i' była spełnialna.

Dowód (dokończenie).

Jeżeli w C_i mamy $k > 3$, to dodajemy $k - 3$ nowych zmiennych $y_{i_2}, \ldots, y_{i_{k-2}}$ i zastępujemy C_i przez

$$C_i' = (x_{i_1} \vee x_{i_2} \vee y_{i_2}) \wedge (\neg y_{i_2} \vee x_{i_3} \vee y_{i_3}) \wedge \cdots \wedge (\neg y_{i_{k-2}} \vee x_{i_{k-1}} \vee x_{i_k}).$$

Jeżeli C_i jest spełnialna, to można dobrać wartościowanie dla $y_{i_2}, \ldots, y_{i_{k-2}}$, aby C_i' była spełnialna.
Jeżeli C_i' jest spełniona dla pewnego wartościowania $x_{i_1}, \ldots, x_{i_k}, y_{i_2}, \ldots, y_{i_{k-2}}$, to $\exists l \in \{1, \ldots, k\}$: $x_{i_l} =$ **prawda**. Istotnie, gdyby $x_{i_1} = x_{i_2} =$ **fałsz**, to z 1. klauzuli w C_i' wynika, że $y_{i_2} =$ **prawda**. Zatem z 2. klauzuli jest $x_{i_3} =$ **prawda** lub $y_{i_3} =$ **prawda**. Przypadek 1. kończy rozumowanie, w przypadku 2. kontynuujemy na kolejnej klauzuli itd.

Dowód (dokończenie).

Jeżeli w C_i mamy $k > 3$, to dodajemy $k - 3$ nowych zmiennych $y_{i_2}, \ldots, y_{i_{k-2}}$ i zastępujemy C_i przez

$$C_i' = (x_{i_1} \vee x_{i_2} \vee y_{i_2}) \wedge (\neg y_{i_2} \vee x_{i_3} \vee y_{i_3}) \wedge \cdots \wedge (\neg y_{i_{k-2}} \vee x_{i_{k-1}} \vee x_{i_k}).$$

Jeżeli C_i jest spełnialna, to można dobrać wartościowanie dla $y_{i_2}, \ldots, y_{i_{k-2}}$, aby C_i' była spełnialna.
Jeżeli C_i' jest spełniona dla pewnego wartościowania $x_{i_1}, \ldots, x_{i_k}, y_{i_2}, \ldots, y_{i_{k-2}}$, to $\exists l \in \{1, \ldots, k\}$: $x_{i_l} =$ **prawda**. Istotnie, gdyby $x_{i_1} = x_{i_2} =$ **fałsz**, to z 1. klauzuli w C_i' wynika, że $y_{i_2} =$ **prawda**. Zatem z 2. klauzuli jest $x_{i_3} =$ **prawda** lub $y_{i_3} =$ **prawda**. Przypadek 1. kończy rozumowanie, w przypadku 2. kontynuujemy na kolejnej klauzuli itd.
Po przekształceniu kolejnych klauzul powstaje równoważna formuła o co najwyżej 3-składnikowych klauzulach.

Dowód (dokończenie).

Jeżeli w C_i mamy $k > 3$, to dodajemy $k-3$ nowych zmiennych $y_{i_2}, \ldots, y_{i_{k-2}}$ i zastępujemy C_i przez

$$C_i' = (x_{i_1} \vee x_{i_2} \vee y_{i_2}) \wedge (\neg y_{i_2} \vee x_{i_3} \vee y_{i_3}) \wedge \cdots \wedge (\neg y_{i_{k-2}} \vee x_{i_{k-1}} \vee x_{i_k}).$$

Jeżeli C_i jest spełnialna, to można dobrać wartościowanie dla $y_{i_2}, \ldots, y_{i_{k-2}}$, aby C_i' była spełnialna.
Jeżeli C_i' jest spełniona dla pewnego wartościowania $x_{i_1}, \ldots, x_{i_k}, y_{i_2}, \ldots, y_{i_{k-2}}$, to $\exists l \in \{1, \ldots, k\}$: $x_{i_l} =$ **prawda**. Istotnie, gdyby $x_{i_1} = x_{i_2} =$ **fałsz**, to z 1. klauzuli w C_i' wynika, że $y_{i_2} =$ **prawda**. Zatem z 2. klauzuli jest $x_{i_3} =$ **prawda** lub $y_{i_3} =$ **prawda**. Przypadek 1. kończy rozumowanie, w przypadku 2. kontynuujemy na kolejnej klauzuli itd.
Po przekształceniu kolejnych klauzul powstaje równoważna formuła o co najwyżej 3-składnikowych klauzulach.
Maszyna Turinga potrzebuje pamięci roboczej tylko na licznik bieżącej klauzuli C_i oraz licznik wygenerowanych nowych zmiennych. Wystarcza pamięć logarytmiczna, więc redukcja dokonuje się w czasie wielomianowym. □

MAX2SAT

Problem (MAX2SAT)

Czy dla danej formuły rachunku zdań zapisanej w koniunktywnej postaci normalnej, w której wszystkie klauzule mają co najwyżej dwa literały, istnieje wartościowanie spełniające przynajmniej k klauzul?

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MAX2SAT

Problem (MAX2SAT)

Czy dla danej formuły rachunku zdań zapisanej w koniunktywnej postaci normalnej, w której wszystkie klauzule mają co najwyżej dwa literały, istnieje wartościowanie spełniające przynajmniej k klauzul?

Fakt

Problem MAX2SAT *jest* **NP**-*zupełny.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Dowód.

Dla formuły $\phi = \bigwedge_{i=1}^{n} C_i$ z przypadku problemu 3SAT, każdą klauzulę $C_i = x_i \vee y_i \vee z_i$ przekształcamy na zbiór 10 klauzul postaci:

$$x_i \wedge y_i \wedge z_i \wedge w_i$$
$$\wedge(\neg x_i \vee \neg y_i) \wedge (\neg y_i \vee \neg z_i) \wedge (\neg x_i \vee \neg z_i) \wedge (x_i \vee w_i) \wedge (y_i \vee \neg w_i) \wedge (z_i \vee \neg w_i).$$

Z powyższych klauzul co najwyżej 7 może być spełnionych, co ma miejsce wtedy i tylko wtedy, gdy C_i jest spełniona.

MAX2SAT

Dowód.

Dla formuły $\phi = \bigwedge_{i=1}^{n} C_i$ z przypadku problemu 3SAT, każdą klauzulę $C_i = x_i \vee y_i \vee z_i$ przekształcamy na zbiór 10 klauzul postaci:

$$x_i \wedge y_i \wedge z_i \wedge w_i$$
$$\wedge(\neg x_i \vee \neg y_i) \wedge (\neg y_i \vee \neg z_i) \wedge (\neg x_i \vee \neg z_i) \wedge (x_i \vee w_i) \wedge (y_i \vee \neg w_i) \wedge (z_i \vee \neg w_i).$$

Z powyższych klauzul co najwyżej 7 może być spełnionych, co ma miejsce wtedy i tylko wtedy, gdy C_i jest spełniona.
Jeżeli w przypadku problemu dla 3SAT jest n klauzul, to zredukowaliśmy ten przypadek do przypadku dla MAX2SAT o $k = 7n$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

MAX2SAT

Dowód.

Dla formuły $\phi = \bigwedge_{i=1}^{n} C_i$ z przypadku problemu 3SAT, każdą klauzulę $C_i = x_i \vee y_i \vee z_i$ przekształcamy na zbiór 10 klauzul postaci:

$$x_i \wedge y_i \wedge z_i \wedge w_i$$
$$\wedge(\neg x_i \vee \neg y_i) \wedge (\neg y_i \vee \neg z_i) \wedge (\neg x_i \vee \neg z_i) \wedge (x_i \vee w_i) \wedge (y_i \vee \neg w_i) \wedge (z_i \vee \neg w_i).$$

Z powyższych klauzul co najwyżej 7 może być spełnionych, co ma miejsce wtedy i tylko wtedy, gdy C_i jest spełniona.
Jeżeli w przypadku problemu dla 3SAT jest n klauzul, to zredukowaliśmy ten przypadek do przypadku dla MAX2SAT o $k = 7n$.
To, że MAX2SAT $\in$ **NP** i że powyższa redukcja może być wykonana w pamięci logarytmicznej, jest oczywiste. □

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy grafowe

Definicja

Graf G to para $G = (V, E)$, gdzie

Problemy grafowe

Definicja

Graf G to para $G = (V, E)$, gdzie

$V \neq \emptyset$ — zbiór wierzchołków,

Problemy grafowe

Definicja

Graf G to para $G = (V, E)$, gdzie

$V \neq \emptyset$ — zbiór wierzchołków,

$E \subseteq \{\{v, w\} : v \in V \wedge w \in V \wedge v \neq w\}$ — zbiór krawędzi.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

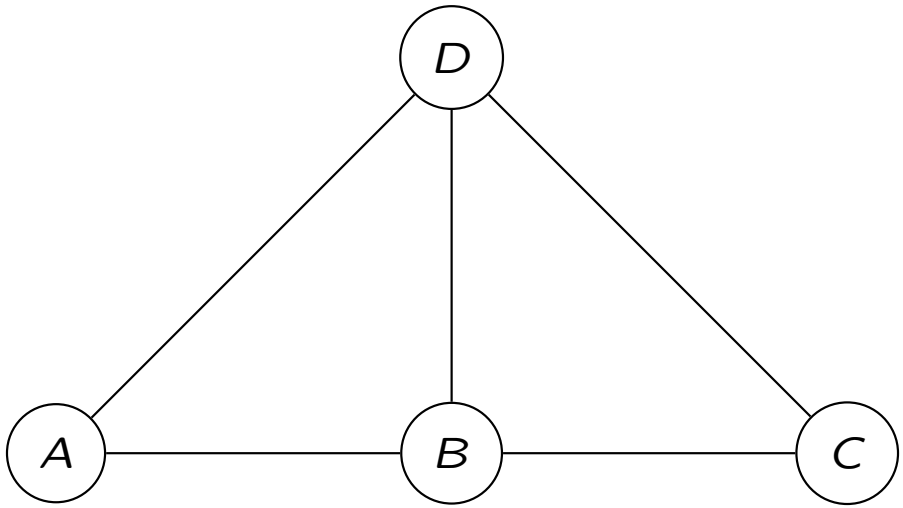
Problemy grafowe

Definicja

Graf G to para $G = (V, E)$, gdzie

$V \neq \emptyset$ — zbiór wierzchołków,

$E \subseteq \{\{v, w\} : v \in V \wedge w \in V \wedge v \neq w\}$ — zbiór krawędzi.



Rysunek: Graf

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Pokrycie wierzchołkowe

Definicja

Pokrycie wierzchołkowe grafu $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$\{v, w\} \in E \implies v \in V' \vee w \in V'.$$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Pokrycie wierzchołkowe

Definicja

Pokrycie wierzchołkowe grafu $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$\{v, w\} \in E \implies v \in V' \vee w \in V'.$$

Problem (NODE COVER)

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Pokrycie wierzchołkowe

Definicja

Pokrycie wierzchołkowe grafu $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$\{v, w\} \in E \implies v \in V' \vee w \in V'.$$

Problem (NODE COVER)

Wejście: $G = (V, E)$, $\mathbb{N} \ni k \leqslant |V|$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Pokrycie wierzchołkowe

Definicja

Pokrycie wierzchołkowe grafu $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$\{v, w\} \in E \implies v \in V' \vee w \in V'.$$

Problem (NODE COVER)

Wejście: $G = (V, E)$, $\mathbb{N} \ni k \leqslant |V|$

Wyjście: **tak**, *jeżeli* G *ma pokrycie wierzchołkowe* V' *i* $|V'| = k$, **nie** *w przeciwnym razie.*

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Pokrycie wierzchołkowe

Definicja

Pokrycie wierzchołkowe grafu $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$\{v, w\} \in E \implies v \in V' \vee w \in V'.$$

Problem (NODE COVER)

Wejście: $G = (V, E)$, $\mathbb{N} \ni k \leqslant |V|$

Wyjście: **tak**, *jeżeli* G *ma pokrycie wierzchołkowe* V' *i* $|V'| = k$, **nie** *w przeciwnym razie.*

Fakt

Problem NODE COVER *jest* **NP***-zupełny.*

Zbiór niezależny i klika

Definicja

Zbiór niezależny w grafie $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$(v \in V' \wedge w \in V') \implies \{v, w\} \notin E.$$

Zbiór niezależny i klika

Definicja

Zbiór niezależny w grafie $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$(v \in V' \wedge w \in V') \implies \{v, w\} \notin E.$$

Problem (INDEPENDENT SET)

Zbiór niezależny i klika

Definicja

Zbiór niezależny w grafie $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$(v \in V' \wedge w \in V') \implies \{v, w\} \notin E.$$

Problem (INDEPENDENT SET)

Wejście: $G = (V, E)$, $\mathbb{N} \ni k \leqslant |V|$

Zbiór niezależny i klika

Definicja

Zbiór niezależny w grafie $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$(v \in V' \wedge w \in V') \implies \{v, w\} \notin E.$$

Problem (INDEPENDENT SET)

Wejście: $G = (V, E)$, $\mathbb{N} \ni k \leqslant |V|$

Wyjście: **tak**, *jeżeli G ma zbiór niezależny V' i $|V'| = k$,*
nie *w przeciwnym razie.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Zbiór niezależny i klika

Definicja

Zbiór niezależny w grafie $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$(v \in V' \wedge w \in V') \implies \{v, w\} \notin E.$$

Problem (INDEPENDENT SET)

Wejście: $G = (V, E)$, $\mathbb{N} \ni k \leqslant |V|$
Wyjście: **tak**, *jeżeli* G *ma zbiór niezależny* V' *i* $|V'| = k$,
nie *w przeciwnym razie.*

Fakt

Problem INDEPENDENT SET *jest* **NP***-zupełny.*

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Zbiór niezależny i klika

Definicja

Klika w grafie $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$(v \in V' \wedge w \in V' \wedge v \neq w) \implies \{v, w\} \in E.$$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Zbiór niezależny i klika

Definicja

Klika w grafie $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$(v \in V' \wedge w \in V' \wedge v \neq w) \implies \{v, w\} \in E.$$

Problem (CLIQUE)

Zbiór niezależny i klika

Definicja

Klika w grafie $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$(v \in V' \wedge w \in V' \wedge v \neq w) \implies \{v, w\} \in E.$$

Problem (CLIQUE)

Wejście: $G = (V, E)$, $\mathbb{N} \ni k \leqslant |V|$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Zbiór niezależny i klika

Definicja

Klika w grafie $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$(v \in V' \wedge w \in V' \wedge v \neq w) \implies \{v, w\} \in E.$$

Problem (CLIQUE)

Wejście: $G = (V, E)$, $\mathbb{N} \ni k \leqslant |V|$
Wyjście: **tak**, *jeżeli* G *ma klikę* V' *i* $|V'| = k$, **nie** *w przeciwnym razie.*

Zbiór niezależny i klika

Definicja

Klika w grafie $G = (V, E)$ to zbiór $V' \subseteq V$ taki, że

$$(v \in V' \wedge w \in V' \wedge v \neq w) \implies \{v, w\} \in E.$$

Problem (CLIQUE)

Wejście: $G = (V, E)$, $\mathbb{N} \ni k \leqslant |V|$
Wyjście: **tak**, *jeżeli* G *ma klikę* V' *i* $|V'| = k$, **nie** *w przeciwnym razie.*

Fakt

Problem CLIQUE *jest* **NP***-zupełny.*

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Cykl Hamiltona i droga Hamiltona

Definicja

Ścieżka łącząca w grafie $G = (V, E)$ wierzchołki $v_1 \in V$ i $v_n \in V$ to ciąg $(v_1, v_2, \ldots, v_n) \in V^n$ taki, że $\forall i \in \{1, \ldots, n-1\}: \{v_i, v_{i+1}\} \in E$.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Cykl Hamiltona i droga Hamiltona

Definicja

Ścieżka łącząca w grafie $G = (V, E)$ wierzchołki $v_1 \in V$ i $v_n \in V$ to ciąg $(v_1, v_2, \ldots, v_n) \in V^n$ taki, że $\forall i \in \{1, \ldots, n-1\}: \{v_i, v_{i+1}\} \in E$.

Definicja

Droga to ścieżka, której wszystkie wierzchołki wewnętrzne są różne: $\forall i, j \in \{1, \ldots, n\}: i \neq j \implies v_i \neq v_j$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Cykl Hamiltona i droga Hamiltona

Definicja

Ścieżka łącząca w grafie $G = (V, E)$ wierzchołki $v_1 \in V$ i $v_n \in V$ to ciąg $(v_1, v_2, \ldots, v_n) \in V^n$ taki, że $\forall i \in \{1, \ldots, n-1\}: \{v_i, v_{i+1}\} \in E$.

Definicja

Droga to ścieżka, której wszystkie wierzchołki wewnętrzne są różne: $\forall i, j \in \{1, \ldots, n\}: i \neq j \implies v_i \neq v_j$.

Definicja

Cykl w grafie $G = (V, E)$ to droga zamknięta $(v_1, v_2, \ldots, v_n, v_1) \in V^{n+1}$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Cykl Hamiltona i droga Hamiltona

Definicja

Cykl Hamiltona w grafie $G = (V, E)$ to cykl $(v_1, v_2, \ldots, v_{|V|}, v_1) \in V^{|V|+1}$ przechodzący przez wszystkie wierzchołki grafu, tzn. $\forall v \in V \; \exists i \in \{1, \ldots, |V|\} : v = v_i$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Cykl Hamiltona i droga Hamiltona

Definicja

Cykl Hamiltona w grafie $G = (V, E)$ to cykl $(v_1, v_2, \ldots, v_{|V|}, v_1) \in V^{|V|+1}$ przechodzący przez wszystkie wierzchołki grafu, tzn. $\forall v \in V \; \exists i \in \{1, \ldots, |V|\} : v = v_i$.

Problem (HAMILTONIAN CYCLE)

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Cykl Hamiltona i droga Hamiltona

Definicja

Cykl Hamiltona w grafie $G = (V, E)$ to cykl $(v_1, v_2, \ldots, v_{|V|}, v_1) \in V^{|V|+1}$ przechodzący przez wszystkie wierzchołki grafu, tzn. $\forall v \in V \; \exists i \in \{1, \ldots, |V|\} : v = v_i$.

Problem (HAMILTONIAN CYCLE)

Wejście: $G = (V, E)$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Cykl Hamiltona i droga Hamiltona

Definicja

Cykl Hamiltona w grafie $G = (V, E)$ to cykl $(v_1, v_2, \ldots, v_{|V|}, v_1) \in V^{|V|+1}$ przechodzący przez wszystkie wierzchołki grafu, tzn. $\forall v \in V \; \exists i \in \{1, \ldots, |V|\} : v = v_i$.

Problem (HAMILTONIAN CYCLE)

Wejście: $G = (V, E)$

Wyjście: **tak**, *jeżeli G ma cykl Hamiltona;* **nie** *w przeciwnym razie.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Cykl Hamiltona i droga Hamiltona

Definicja

Cykl Hamiltona w grafie $G = (V, E)$ to cykl $(v_1, v_2, \ldots, v_{|V|}, v_1) \in V^{|V|+1}$ przechodzący przez wszystkie wierzchołki grafu, tzn. $\forall v \in V \; \exists i \in \{1, \ldots, |V|\} : v = v_i$.

Problem (HAMILTONIAN CYCLE)

Wejście: $G = (V, E)$
Wyjście: **tak**, *jeżeli G ma cykl Hamiltona;* **nie** *w przeciwnym razie.*

Fakt

Problem HAMILTONIAN CYCLE *jest* **NP**-*zupełny.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Cykl Hamiltona i droga Hamiltona

Definicja

Droga Hamiltona w grafie $G = (V, E)$ to droga $(v_1, v_2, \ldots, v_{|V|}) \in V^{|V|}$ przechodząca przez wszystkie wierzchołki grafu, tzn. $\forall v \in V \; \exists i \in \{1, \ldots, |V|\} : v = v_i$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Cykl Hamiltona i droga Hamiltona

Definicja

Droga Hamiltona w grafie $G = (V, E)$ to droga $(v_1, v_2, \ldots, v_{|V|}) \in V^{|V|}$ przechodząca przez wszystkie wierzchołki grafu, tzn. $\forall v \in V \; \exists i \in \{1, \ldots, |V|\} : v = v_i$.

Problem (HAMILTONIAN PATH)

Cykl Hamiltona i droga Hamiltona

Definicja

Droga Hamiltona w grafie $G = (V, E)$ to droga $(v_1, v_2, \ldots, v_{|V|}) \in V^{|V|}$ przechodząca przez wszystkie wierzchołki grafu, tzn. $\forall v \in V \; \exists i \in \{1, \ldots, |V|\} : v = v_i$.

Problem (HAMILTONIAN PATH)

Wejście: $G = (V, E)$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Cykl Hamiltona i droga Hamiltona

Definicja

Droga Hamiltona w grafie $G = (V, E)$ to droga $(v_1, v_2, \ldots, v_{|V|}) \in V^{|V|}$ przechodząca przez wszystkie wierzchołki grafu, tzn. $\forall v \in V \; \exists i \in \{1, \ldots, |V|\} : v = v_i$.

Problem (HAMILTONIAN PATH)

Wejście: $G = (V, E)$

Wyjście: **tak**, *jeżeli G ma drogę Hamiltona;* **nie** *w przeciwnym razie.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Cykl Hamiltona i droga Hamiltona

Definicja

Droga Hamiltona w grafie $G = (V, E)$ to droga $(v_1, v_2, \ldots, v_{|V|}) \in V^{|V|}$ przechodząca przez wszystkie wierzchołki grafu, tzn. $\forall v \in V \; \exists i \in \{1, \ldots, |V|\} : v = v_i$.

Problem (HAMILTONIAN PATH)

Wejście: $G = (V, E)$
Wyjście: **tak**, *jeżeli G ma drogę Hamiltona;* **nie** *w przeciwnym razie.*

Fakt

Problem HAMILTONIAN PATH *jest* **NP***-zupełny.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy na zbiorach i liczbach

3. Problemy na zbiorach i liczbach
   - Skojarzenie trójdzielne
   - Problemy dot. pokryć
   - Problemy liczbowe

Skojarzenie trójdzielne

Problem (TRIPARTITE MATCHING)

Skojarzenie trójdzielne

Problem (TRIPARTITE MATCHING)

Wejście: *zbiory skończone* X, Y, Z, *parami rozłączne oraz relacja* $R \subset X \times Y \times Z$

Skojarzenie trójdzielne

Problem (TRIPARTITE MATCHING)

Wejście: *zbiory skończone* X, Y, Z, *parami rozłączne oraz relacja* $R \subset X \times Y \times Z$

Wyjście: **tak**, *jeżeli istnieje skojarzenie:*
$\exists M \subseteq R\colon (x, y, z) \in M \wedge (x', y', z') \in M \wedge (x, y, z) \neq (x', y', z') \implies x \neq x' \wedge y \neq y' \wedge z \neq z'$; **nie** *w przeciwnym razie.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Skojarzenie trójdzielne

Problem (TRIPARTITE MATCHING)

Wejście: *zbiory skończone* X, Y, Z, *parami rozłączne oraz relacja* $R \subset X \times Y \times Z$

Wyjście: **tak**, *jeżeli istnieje skojarzenie:*
$\exists M \subseteq R\colon (x, y, z) \in M \wedge (x', y', z') \in M \wedge (x, y, z) \neq (x', y', z') \implies x \neq x' \wedge y \neq y' \wedge z \neq z'$; **nie** *w przeciwnym razie.*

Fakt

Problem TRIPARTITE MATCHING *jest* **NP***-zupełny.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Skojarzenie trójdzielne

Problem (TRIPARTITE MATCHING)

Wejście: *zbiory skończone* X, Y, Z, *parami rozłączne oraz relacja* $R \subset X \times Y \times Z$

Wyjście: **tak**, *jeżeli istnieje skojarzenie:*
$\exists M \subseteq R\colon (x,y,z) \in M \wedge (x',y',z') \in M \wedge (x,y,z) \neq (x',y',z') \implies x \neq x' \wedge y \neq y' \wedge z \neq z'$; **nie** *w przeciwnym razie.*

Fakt

Problem TRIPARTITE MATCHING *jest* **NP**-*zupełny.*

Idea dowodu.

Redukcja z 3SAT. □

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ Uniwersytet Łódzki

Skojarzenie trójdzielne

Dla porównania:

Problem (BIPARTITE MATCHING)

Skojarzenie trójdzielne

Dla porównania:

Problem (BIPARTITE MATCHING)

Wejście: *zbiory skończone* X, Y, *rozłączne oraz relacja* $R \subset X \times Y$

Skojarzenie trójdzielne

Dla porównania:

Problem (BIPARTITE MATCHING)

Wejście: *zbiory skończone* X, Y, *rozłączne oraz relacja* $R \subset X \times Y$

Wyjście: **tak**, *jeżeli istnieje skojarzenie:* $\exists M \subseteq R\colon (x,y) \in M \wedge (x',y') \in M \wedge (x,y) \neq (x',y') \implies x \neq x' \wedge y \neq y'$; **nie** *w przeciwnym razie.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Skojarzenie trójdzielne

Dla porównania:

Problem (BIPARTITE MATCHING)

Wejście: *zbiory skończone* X, Y, *rozłączne oraz relacja* $R \subset X \times Y$

Wyjście: **tak**, *jeżeli istnieje skojarzenie:* $\exists M \subseteq R\colon (x,y) \in M \wedge (x',y') \in M \wedge (x,y) \neq (x',y') \implies x \neq x' \wedge y \neq y'$; **nie** *w przeciwnym razie.*

Fakt

BIPARTITE MATCHING $\in \mathbf{P}$.

Skojarzenie trójdzielne

Dla porównania:

Problem (BIPARTITE MATCHING)

Wejście: *zbiory skończone* X, Y, *rozłączne oraz relacja* $R \subset X \times Y$

Wyjście: **tak**, *jeżeli istnieje skojarzenie:* $\exists M \subseteq R\colon (x,y) \in M \wedge (x',y') \in M \wedge (x,y) \neq (x',y') \implies x \neq x' \wedge y \neq y'$; **nie** *w przeciwnym razie.*

Fakt

BIPARTITE MATCHING $\in \mathbf{P}$.

Rozwiązanie

Skojarzenie dla BIPARTITE MATCHING *można wyznaczyć za pomocą algorytmu Hopcrofta–Karpa, który wymaga czasu* $O(|R|\sqrt{|X|+|Y|})$.

Problemy dot. pokryć

Problem (EXACT COVER BY 3-SETS)

Problemy dot. pokryć

Problem (EXACT COVER BY 3-SETS)

Wejście: *zbiór* X *taki, że* $\exists n \in \mathbb{N}\colon |X| = 3n$, *rodzina podzbiorów* $\mathcal{S} \subset \mathcal{P}(X)$ *taka, że* $\forall U \in \mathcal{S}\colon |U| = 3$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy dot. pokryć

Problem (EXACT COVER BY 3-SETS)

Wejście: *zbiór* X *taki, że* $\exists n \in \mathbb{N}\colon |X| = 3n$, *rodzina podzbiorów* $\mathcal{S} \subset \mathcal{P}(X)$ *taka, że* $\forall U \in \mathcal{S}\colon |U| = 3$

Wyjście: **tak**, *jeżeli* $\exists \mathcal{C} \subseteq \mathcal{S}\colon (\bigcup_{U \in \mathcal{C}} = X) \wedge (\forall x \in X\colon x \in U \in \mathcal{C} \wedge x \in U' \in \mathcal{C} \implies U = U')$; **nie** *w przeciwnym razie.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy dot. pokryć

Problem (EXACT COVER BY 3-SETS)

Wejście: *zbiór* X *taki, że* $\exists n \in \mathbb{N}\colon |X| = 3n$, *rodzina podzbiorów* $\mathcal{S} \subset \mathcal{P}(X)$ *taka, że* $\forall U \in \mathcal{S}\colon |U| = 3$

Wyjście: **tak**, *jeżeli* $\exists \mathcal{C} \subseteq \mathcal{S}\colon (\bigcup_{U \in \mathcal{C}} = X) \wedge (\forall x \in X\colon x \in U \in \mathcal{C} \wedge x \in U' \in \mathcal{C} \implies U = U')$; **nie** *w przeciwnym razie.*

Fakt

Problem EXACT COVER BY 3-SETS *jest* **NP***-zupełny.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy dot. pokryć

Problem (SET COVERING)

Problemy dot. pokryć

Problem (SET COVERING)

Wejście: *zbiór* X, *rodzina* $\mathcal{S} \subset \mathcal{P}(X)$, *liczba* $\mathbb{N} \ni k \leqslant |\mathcal{S}|$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy dot. pokryć

Problem (SET COVERING)

Wejście: *zbiór* X*, rodzina* $\mathcal{S} \subset \mathcal{P}(X)$*, liczba* $\mathbb{N} \ni k \leqslant |\mathcal{S}|$
Wyjście: **tak**, *jeżeli* $\exists \mathcal{C} \subseteq \mathcal{S}: \bigcup_{U \in \mathcal{C}} = X \wedge |\mathcal{C}| = k$; **nie** *w przeciwnym razie.*

Problemy dot. pokryć

Problem (SET COVERING)

Wejście: *zbiór* X, *rodzina* $\mathcal{S} \subset \mathcal{P}(X)$, *liczba* $\mathbb{N} \ni k \leqslant |\mathcal{S}|$
Wyjście: **tak**, *jeżeli* $\exists \mathcal{C} \subseteq \mathcal{S}: \bigcup_{U \in \mathcal{C}} = X \wedge |\mathcal{C}| = k$; **nie** *w przeciwnym razie.*

Fakt

Problem SET COVERING *jest* **NP**-*zupełny.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy liczbowe

Problem (SUBSET SUM)

Problemy liczbowe

Problem (SUBSET SUM)

Wejście: *skończony zbiór* A, *funkcja* $s\colon A \to \mathbb{N} \cup \{0\}$, *liczba* $B \in \mathbb{N} \cup \{0\}$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy liczbowe

Problem (SUBSET SUM)

Wejście: *skończony zbiór* A, *funkcja* $s\colon A \to \mathbb{N} \cup \{0\}$, *liczba* $B \in \mathbb{N} \cup \{0\}$

Wyjście: **tak**, *jeżeli* $\exists A' \subseteq A\colon \sum_{a \in A'} s(a) = B$; **nie** *w przeciwnym razie.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy liczbowe

Problem (SUBSET SUM)

Wejście: *skończony zbiór* A, *funkcja* $s\colon A \to \mathbb{N} \cup \{0\}$, *liczba* $B \in \mathbb{N} \cup \{0\}$

Wyjście: **tak**, *jeżeli* $\exists A' \subseteq A\colon \sum_{a \in A'} s(a) = B$; **nie** *w przeciwnym razie.*

Fakt

Problem SUBSET SUM *jest* **NP**-*zupełny.*

Problemy liczbowe

Problem (SUBSET SUM)

Wejście: *skończony zbiór* A, *funkcja* $s\colon A \to \mathbb{N} \cup \{0\}$, *liczba* $B \in \mathbb{N} \cup \{0\}$

Wyjście: **tak**, *jeżeli* $\exists A' \subseteq A\colon \sum_{a \in A'} s(a) = B$; **nie** *w przeciwnym razie.*

Fakt

Problem SUBSET SUM *jest* **NP**-*zupełny.*

Idea dowodu.

Redukcja z EXACT COVER BY 3-SETS. □

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy liczbowe

Problem (PARTITION)

Problemy liczbowe

Problem (PARTITION)

Wejście: *skończony zbiór* A, *funkcja* $s\colon A \to \mathbb{N} \setminus \{0\}$

Problemy liczbowe

Problem (PARTITION)

Wejście: *skończony zbiór* A, *funkcja* $s\colon A \to \mathbb{N} \setminus \{0\}$
Wyjście: **tak**, *jeżeli* $\exists A' \subseteq A\colon \sum_{a\in A'} s(a) = \sum_{a\in A\setminus A'} s(a)$;
nie *w przeciwnym razie.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy liczbowe

Problem (PARTITION)

Wejście: *skończony zbiór* A, *funkcja* $s\colon A \to \mathbb{N} \setminus \{0\}$
Wyjście: **tak**, *jeżeli* $\exists A' \subseteq A\colon \sum_{a\in A'} s(a) = \sum_{a\in A\setminus A'} s(a)$;
nie *w przeciwnym razie.*

Fakt

Problem PARTITION *jest* **NP**-*zupełny.*

Problemy liczbowe

Problem (KNAPSACK)

Problemy liczbowe

Problem (KNAPSACK)

Wejście: *skończony zbiór* A, *funkcje* $s\colon A \to \mathbb{N} \cup \{0\}$, $v\colon A \to \mathbb{N} \cup \{0\}$, *liczby* $B \in \mathbb{N} \cup \{0\}$, $K \in \mathbb{N} \cup \{0\}$

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Problemy liczbowe

Problem (KNAPSACK)

Wejście: *skończony zbiór* A, *funkcje* $s\colon A \to \mathbb{N} \cup \{0\}$, $v\colon A \to \mathbb{N} \cup \{0\}$, *liczby* $B \in \mathbb{N} \cup \{0\}$, $K \in \mathbb{N} \cup \{0\}$

Wyjście: **tak**, *jeżeli* $\exists A' \subseteq A\colon \sum_{a \in A'} s(a) \leqslant B \wedge \sum_{a \in A'} v(a) \geqslant K$; **nie** *w przeciwnym razie.*

Problemy liczbowe

Problem (KNAPSACK)

Wejście: *skończony zbiór* A, *funkcje* $s\colon A \to \mathbb{N} \cup \{0\}$, $v\colon A \to \mathbb{N} \cup \{0\}$, *liczby* $B \in \mathbb{N} \cup \{0\}$, $K \in \mathbb{N} \cup \{0\}$

Wyjście: **tak**, *jeżeli* $\exists A' \subseteq A\colon \sum_{a \in A'} s(a) \leqslant B \wedge \sum_{a \in A'} v(a) \geqslant K$; **nie** *w przeciwnym razie.*

Fakt

Problem KNAPSACK *jest* **NP**-*zupełny.*

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki