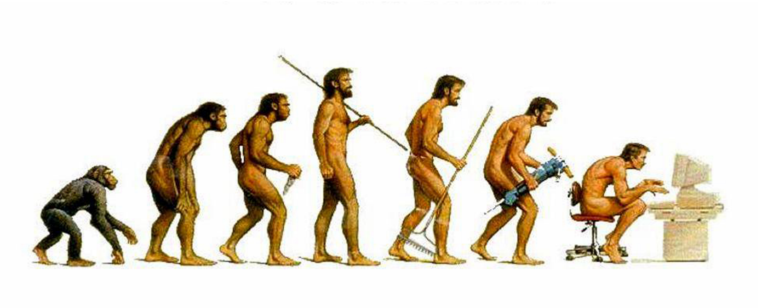


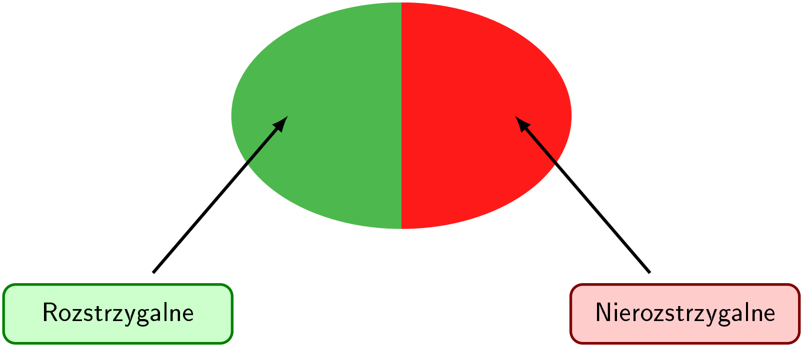
Teoria Obliczalności

Marcin Piątkowski

Wykład 9

Czasowa złożoność obliczeniowa

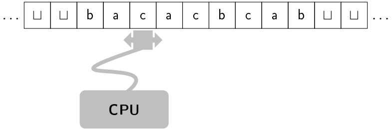
Podział problemów obliczeniowych



Wybór modelu

! Przy założeniu nieograniczoności dostępnych zasobów (czas oraz pamięć) wszystkie poznane do tej pory modele obliczeń są równoważne.

	5	20	8	4	13	0	0	0	0	0	0	0	…
L	0	1	2	3	4	5	5	6	7	8	9	10	



$$
\begin{cases}
h(\overline{x}, 0) = f(\overline{x}) \\
h(\overline{x}, y+1) = g\Big(\overline{x}, y, h(\overline{x}, y)\Big)
\end{cases}
$$

Problem

$$PATH = \Big\{(G, v_1, v_2) : \text{ graf } G \text{ zawiera ścieżkę } v_1 \rightsquigarrow v_2\Big\}$$

Problem

$$PATH = \Big\{(G, v_1, v_2) : \text{ graf } G \text{ zawiera ścieżkę } v_1 \rightsquigarrow v_2\Big\}$$

Input: $(\mathtt{G},\ \mathtt{v}_1,\ \mathtt{v}_2)$

if (*istnieje ścieżka* $\mathtt{v}_1 \rightsquigarrow \mathtt{v}_2$)

    return `true;`

else

    return `false`

Poziom szczegółowości algorytmu

Problem

$$PATH = \Big\{(G, v_1, v_2) : \text{ graf } G \text{ zawiera ścieżkę } v_1 \rightsquigarrow v_2\Big\}$$

Input: $(\mathtt{G},\ \mathtt{v}_1,\ \mathtt{v}_2)$

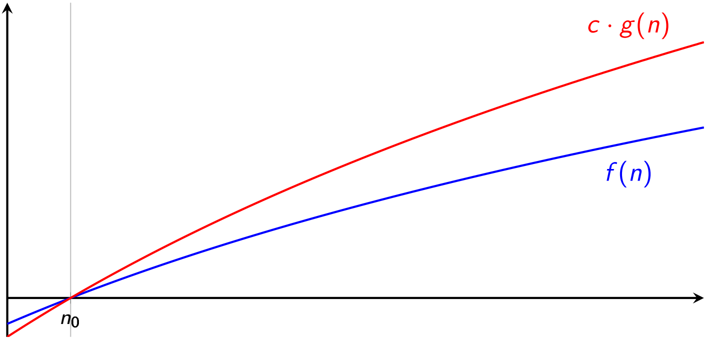
```
kolejka.dodaj(v₁)
while (kolejka ≠ ∅)
    v → kolejka.usun()
    foreach (s - nieodwiedzony sąsiad v)
        if (s == v₂)
            return true
        zaznacz s jako odwiedzony
        kolejka.dodaj(s)
return false
```

Złożoność czasowa

Złożonością czasową (czasem działania) **deterministycznej** maszyny Turinga M nazywamy funkcję $f : \boldsymbol{N} \longrightarrow \boldsymbol{N}$, gdzie $f(n)$ jest równa **największej** liczbie kroków wykonywanych przez maszynę M dla **dowolnego** słowa długości n

Asymptotyczne ograniczenie górne

Niech $f, g : \mathbb{N} \longrightarrow \mathbb{R}^+$. Powiemy, że $f(n) = O(g(n))$ jeśli istnieją dodatnie stałe c oraz n_0 takie, że $\forall_{n \geq n_0}\ f(n) \leq c \cdot g(n)$.



Asymptotyczne ograniczenie górne

Niech $f, g : \boldsymbol{N} \longrightarrow \boldsymbol{R}^+$. Powiemy, że $f(n) = O\big(g(n)\big)$ jeśli istnieją dodatnie stałe c oraz n_0 takie, że $\forall_{n \geq n_0}\ f(n) \leq c \cdot g(n)$.

- ☞ $f(n) = 2n^2 + 2n - 1$ jest $O(n^2)$ ale nie jest $O(n)$
- ☞ $f(n) = 2n - 1$ jest $O(n^2)$, $O(n)$ ale nie $O(\log n)$
- ☞ $f(n) = \log_2 n$ jest $O(n)$, $O(\log n)$
- ☞ $f(n) = 4$ jest $O(1)$

Asymptotyczne ograniczenie górne

Niech $f, g : \boldsymbol{N} \longrightarrow \boldsymbol{R}^{+}$. Powiemy, że $f(n) = O\big(g(n)\big)$ jeśli istnieją dodatnie stałe c oraz n_0 takie, że $\forall_{n \geq n_0}\ f(n) \leq c \cdot g(n)$.

- ☞ $f(n) = 2n^2 + 2n - 1$ jest $O(n^2)$ ale nie jest $O(n)$
- ☞ $f(n) = 2n - 1$ jest $O(n^2)$, $O(n)$ ale nie $O(\log n)$
- ☞ $f(n) = \log_2 n$ jest $O(n)$, $O(\log n)$
- ☞ $f(n) = 4$ jest $O(1)$

! $$\log_a \mathbf{x} = \frac{1}{\log_b a} \cdot \log_b \mathbf{x}$$

Asymptotyczne ograniczenie górne

Niech $f, g : \boldsymbol{N} \longrightarrow \boldsymbol{R}^{+}$. Powiemy, że $f(n) = o\Big(g(n)\Big)$ jeśli

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Równoważnie

$$\forall_{c>0}\, \exists_{n_0}\, \forall_{n \geq n_0}\, f(n) \leq c \cdot g(n)$$

- ☞ $f(n) = n^2$ jest $o(n^3)$
- ☞ $f(n) = \sqrt{n}$ jest $o(n)$
- ☞ $f(n) = 3n$ nie jest $o(n)$

Problem

$$L = \left\{ a^k b^k : \; k \geq 0 \right\}$$

? Ile operacji musi wykonać maszyna Turinga rozstrzygająca język L?

Maszyna z jedną taśmą

1. Zweryfikuj czy dane na taśmie są postaci $a \ldots ab \ldots b$
2. Usuwaj naprzemiennie a z początku oraz b z końca słowa
3. **Zaakceptuj** jeśli taśma jest pusta
4. **Odrzuć** jeśli taśma zawiera wyłącznie a lub wyłącznie b

Maszyna z jedną taśmą

1. Zweryfikuj czy dane na taśmie są postaci $a \dots ab \dots b$ $O(n)$
2. Usuwaj naprzemiennie a z początku oraz b z końca słowa $O(n^2)$
3. **Zaakceptuj** jeśli taśma jest pusta
4. **Odrzuć** jeśli taśma zawiera wyłącznie a lub wyłącznie b

Całkowity czas działania: $O(n) + O(n^2) = O(n^2)$

Maszyna z jedną taśmą

1. Zweryfikuj czy dane na taśmie są postaci $a \ldots ab \ldots b$
2. Dopóki taśma nie jest pusta:
   - Zweryfikuj czy liczba znaków na taśmie jest parzysta
   - Usuń co drugie wystąpienie a oraz co drugie wystąpienie b
3. **Zaakceptuj** jeśli taśma jest pusta

Maszyna z jedną taśmą

1. Zweryfikuj czy dane na taśmie są postaci $a \dots ab \dots b$ $O(n)$
2. Dopóki taśma nie jest pusta: $\Big\} O(\log n)$
   - Zweryfikuj czy liczba znaków na taśmie jest parzysta $O(n)$
   - Usuń co drugie wystąpienie a oraz co drugie wystąpienie b $O(n)$
3. **Zaakceptuj** jeśli taśma jest pusta

Całkowity czas działania: $O(n) + O(n \log n) = O(n \log n)$

Maszyna z dwoma taśmami

1. Skopiuj początkowe wystąpienia a na drugą taśmę
2. Dla każdego b na pierwszej taśmie usuwaj a z drugiej
3. **Zaakceptuj** jeśli po osiągnięciu prawego końca słowa wejściowego taśma druga jest pusta
4. **Odrzuć** jeśli:
   - po osiągnięciu końca słowa na pierwszej taśmie druga nie jest pusta
   - druga taśma będzie pusta przed osiągnięciem końca słowa na pierwszej taśmie
   - za ostatnim b znajduje się a

Maszyna z dwoma taśmami

1. Skopiuj początkowe wystąpienia a na drugą taśmę
2. Dla każdego b na pierwszej taśmie usuwaj a z drugiej

$O(n)$

3. **Zaakceptuj** jeśli po osiągnięciu prawego końca słowa wejściowego taśma druga jest pusta
4. **Odrzuć** jeśli:
   - po osiągnięciu końca słowa na pierwszej taśmie druga nie jest pusta
   - druga taśma będzie pusta przed osiągnięciem końca słowa na pierwszej taśmie
   - za ostatnim b znajduje się a

Całkowity czas działania: $O(n)$

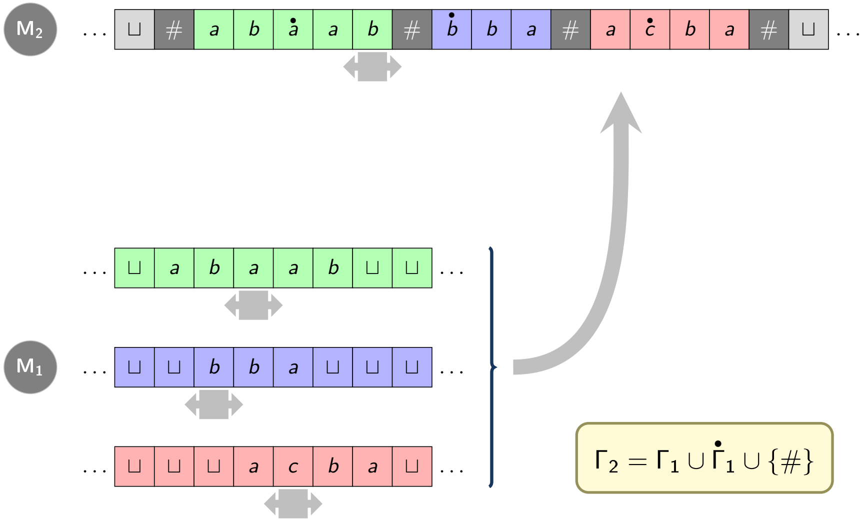
Złożoność algorytmu vs złożoność problemu

Złożoność algorytmu $\neq$ Złożoność problemu

Twierdzenie

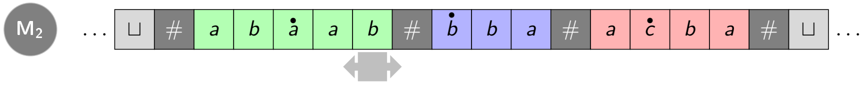
Niech $f : \boldsymbol{N} \to \boldsymbol{R}$ będzie funkcją spełniającą warunek $f(n) \geq n$. Wówczas dla dowolnej k-taśmowej maszyny Turinga M_1 o złożoności czasowej $O\big(f(n)\big)$ istnieje **równoważna jej** jednotaśmowa maszyna M_2 działająca w czasie $O\big(f^2(n)\big)$.

Maszyny wielotaśmowe



$$\Gamma_2 = \Gamma_1 \cup \dot{\Gamma}_1 \cup \{\#\}$$

Maszyny wielotaśmowe



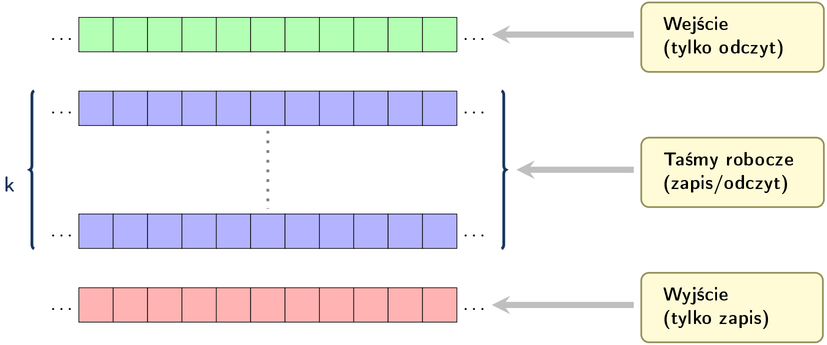
Założenie Maszyna M_1 działa w czasie $O\big(f(n)\big)$

- W czasie $O\big(f(n)\big)$ maszyna M_1 może zapisać co najwyżej $O\big(f(n)\big)$ komórek taśmy
- Symulacja pojedynczego kroku $M_1 \implies O\big(f(n)\big)$
- Symulacja $M_1 \implies O\big(f(n)\big)$ kroków do wykonania

Wniosek Maszyna M_2 działa w czasie $O\big(f^2(n)\big)$

Model

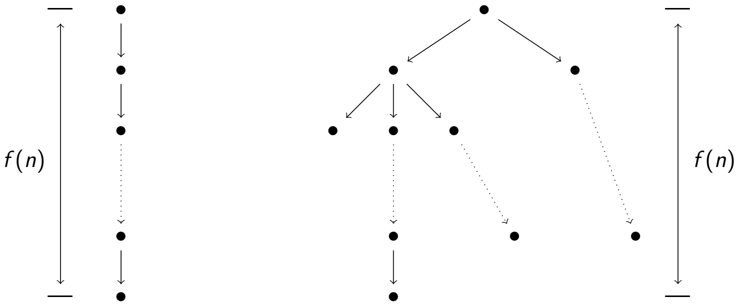
Maszyna (k+2)-taśmowa



? Uzasadnij wielomianową równoważność maszyn Turinga oraz maszyn licznikowych

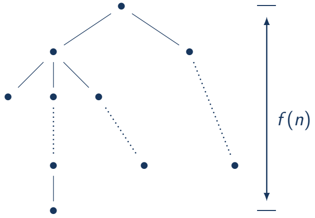
Niedeterministyczna złożoność czasowa

Złożonością czasową (czasem działania) **niedeterministycznej** maszyny Turinga M nazywamy funkcję $f : \mathbb{N} \longrightarrow \mathbb{N}$, gdzie $f(n)$ jest równa **największej** liczbie kroków wykonywanych przez maszynę M na **dowolnej** ścieżce obliczeń dla **dowolnego** słowa długości n



Twierdzenie

Niech $f : \boldsymbol{N} \longrightarrow \boldsymbol{R}$ będzie funkcją spełniającą warunek $f(n) \geq n$. Wówczas dla każdej jednotaśmowej **niedeterministycznej** maszyny Turinga M_1 o złożoności czasowej $O\big(f(n)\big)$ istnieje **równoważna jej** jednotaśmowa **deterministyczna** maszyna Turinga M_2 działająca w czasie $2^{O(f(n))}$.

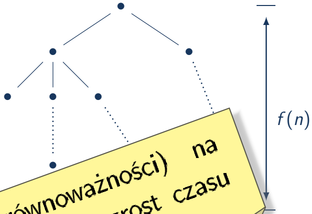


Założenie Maszyna M_1 działa w czasie $O\big(f(n)\big)$

- ☞ Symulacja działania $M_1 \Longrightarrow$ przeglądanie (wszerz) jej drzewa obliczeń
- ☞ Każda ścieżka w drzewie obliczeń M_1 ma długość co najwyżej $O\big(f(n)\big)$
- ☞ Każdy węzeł w drzewie obliczeń M_1 ma co najwyżej b potomków
- ☞ Maksymalna liczba węzłów w drzewie obliczeń M_1 jest rzędu $O\big(b^{f(n)}\big)$

Wniosek Maszyna M_2 działa w czasie $O\big(f(n)\big) \cdot O\big(b^{f(n)}\big) \;=\; 2^{O(f(n))}$

Determinizm vs niedeterminizm



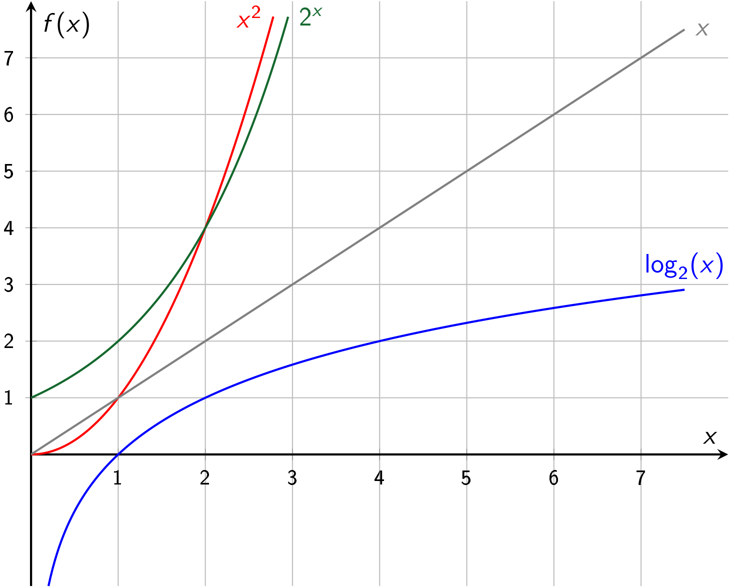
Założenie Maszyna M_1 działa w czasie $O(\ldots$

- Symulacja działania … jej drzewa obliczeń
- Każ… M_1 ma długość co najwyżej $O(f(n))$
- …zewie obliczeń M_1 ma co najwyżej b potomków
- Maksymalna liczba węzłów w drzewie obliczeń M_1 jest rzędu $O(b^{f(n)})$

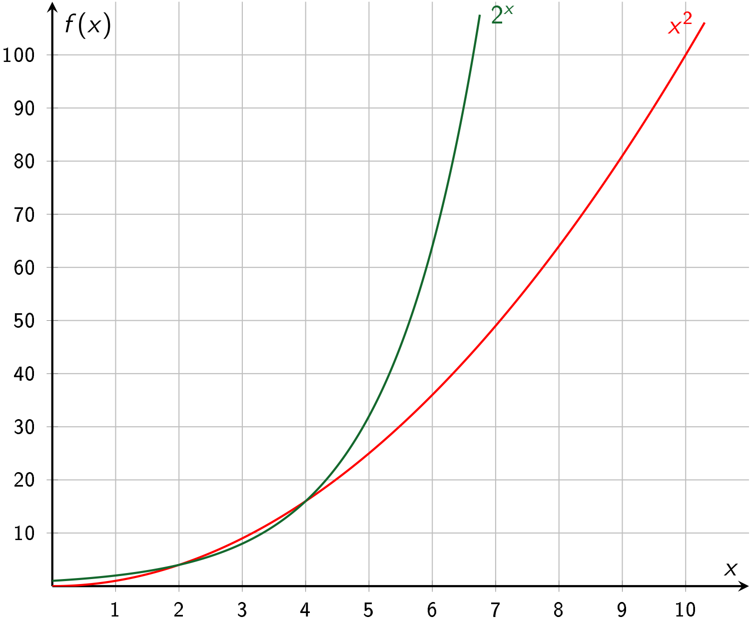
Wniosek Maszyna M_2 działa w czasie $O(f(n)) \cdot O(b^{f(n)}) = 2^{O(f(n))}$

! Zmiana maszyny **trzytaśmowej** (dowód równoważności) na **jednotaśmową** powoduje maksymalnie kwadratowy wzrost czasu obliczeń maszyny symulującej

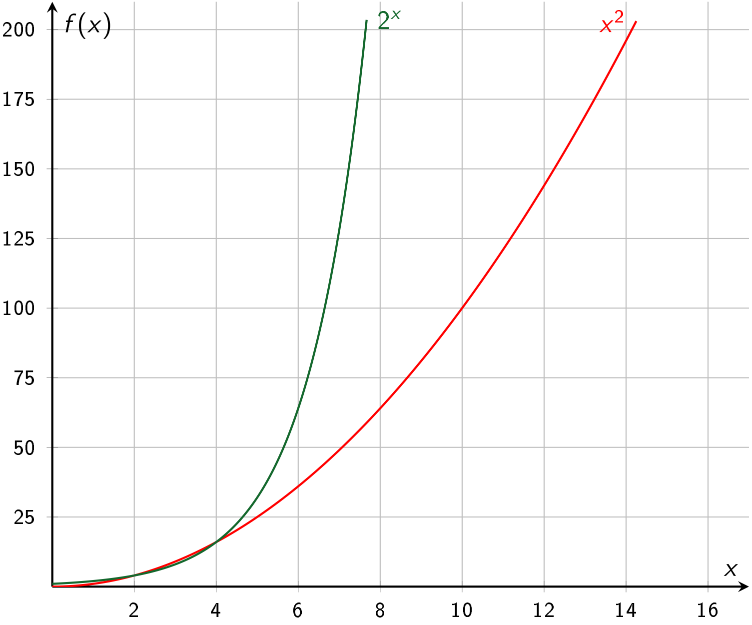
Skalowalność



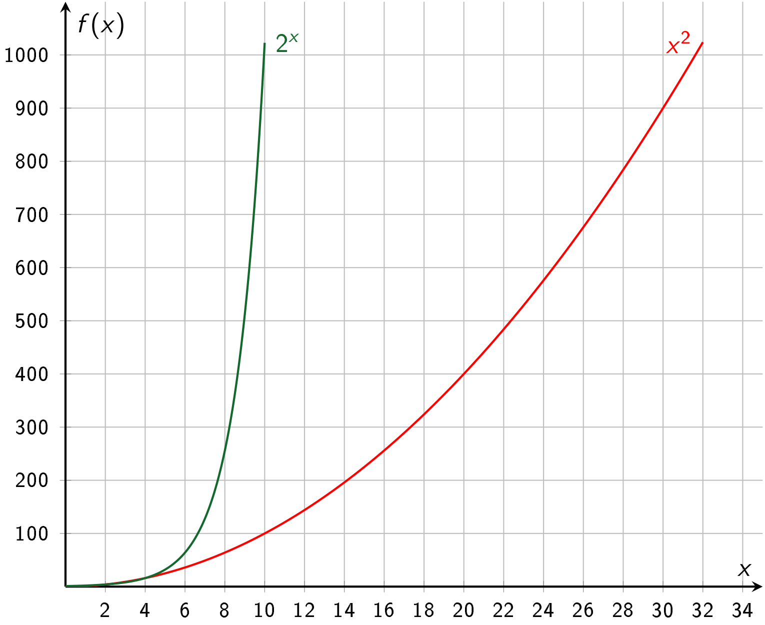
Skalowalność



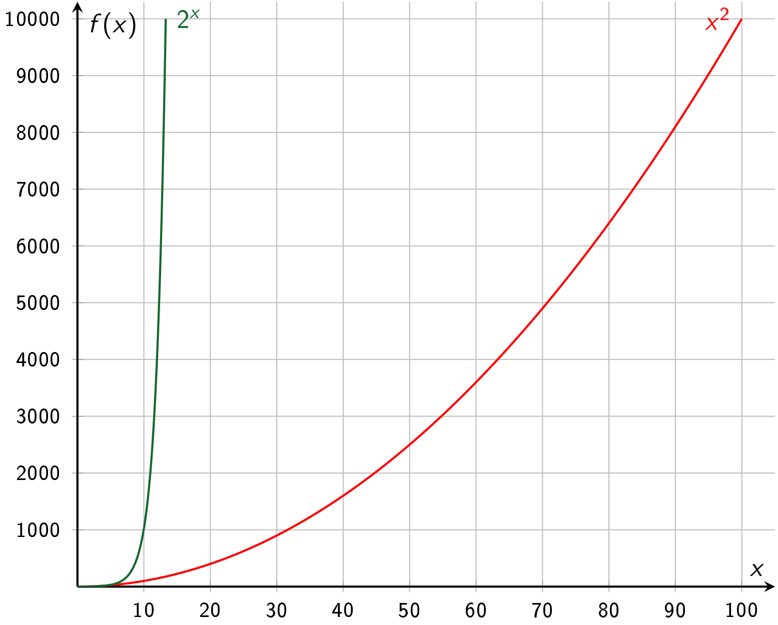
Skalowalność



Skalowalność



Skalowalność



Klasy czasowej złożoności obliczeniowej

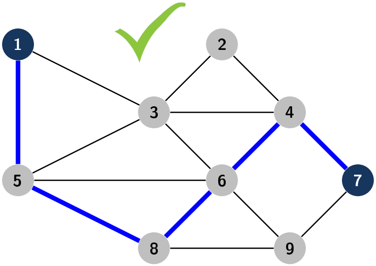
DTIME(f(n)) — zbiór wszystkich problemów rozstrzyganych w czasie $O(f(n))$ przez **deterministyczne** maszyny Turinga

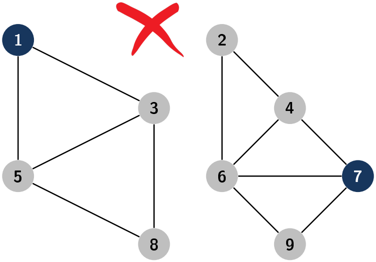
NTIME(f(n)) — zbiór wszystkich problemów rozstrzyganych w czasie $O(f(n))$ przez **niedeterministyczne** maszyne Turinga

PTIME (P)

$$P = \bigcup_k DTIME(n^k)$$

$$PATH = \left\{(G, v_1, v_2) : G \text{ zawiera ścieżkę } v_1 \rightsquigarrow v_2\right\}$$





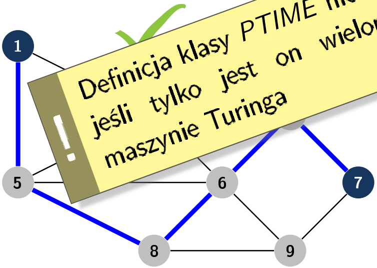
PTIME (P)

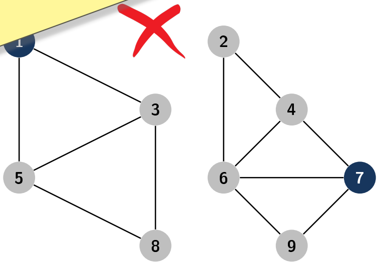
$$P = \bigcup_k DTIME(n^k)$$

$PATH = \{(G, v_1, v_2) : G$ zawier

Definicja klasy PTIME nie zależy od wyboru modelu obliczeniowego, jeśli tylko jest on wielomianowo równoważny deterministycznej maszynie Turinga

!





EXPTIME

$$EXPTIME = \bigcup_k DTIME(2^{n^k})$$

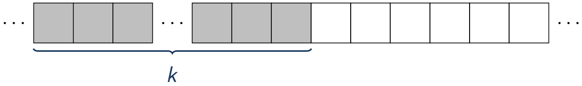
$$HALT_K = \left\{(M, k) : \forall_w \, M \text{ zatrzymuje się po co najwyżej } k \text{ krokach}\right\}$$

Klasa EXPTIME

EXPTIME

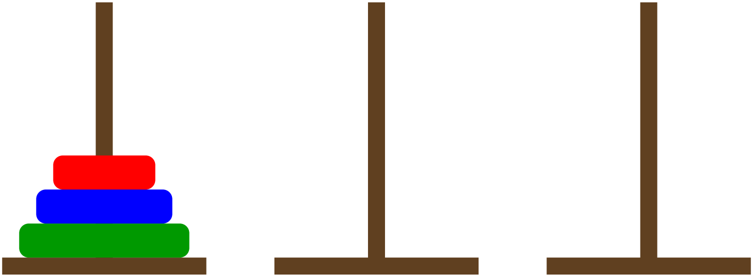
$$EXPTIME \;=\; \bigcup_k \; DTIME(2^{n^k})$$

$$HALT_K = \Big\{(M, k) : \ \forall_w \ M \,\text{zatrzymuje się po co najwyżej}\, k \,\text{krokach}\Big\}$$

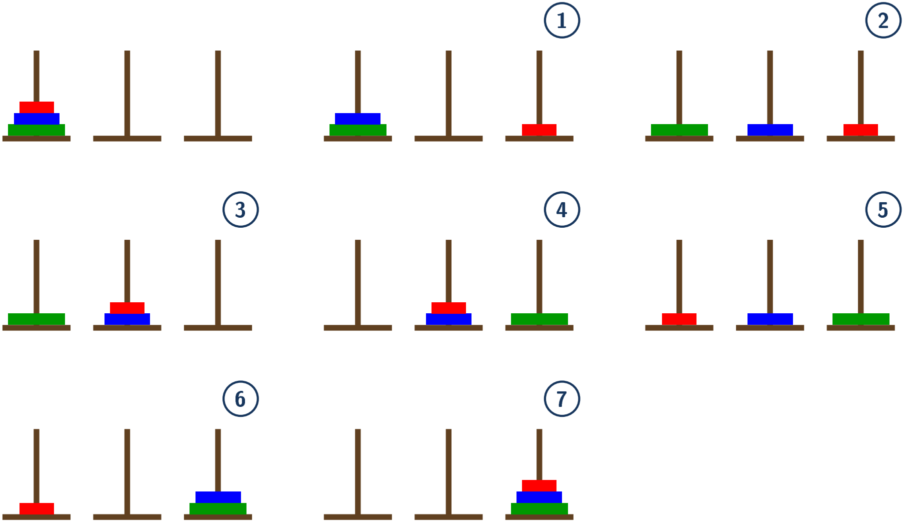


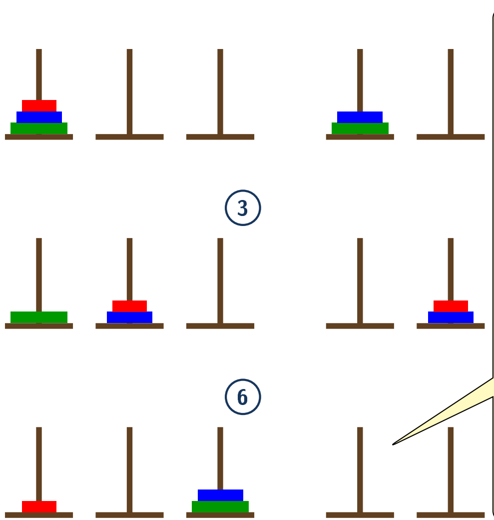
- ☞ Maszyna M przeczytać tylko k początkowych komórek taśmy
- ☞ Symulacja działania M na każdym możliwym wejściu długości k

Wieże Hanoi



Wieże Hanoi





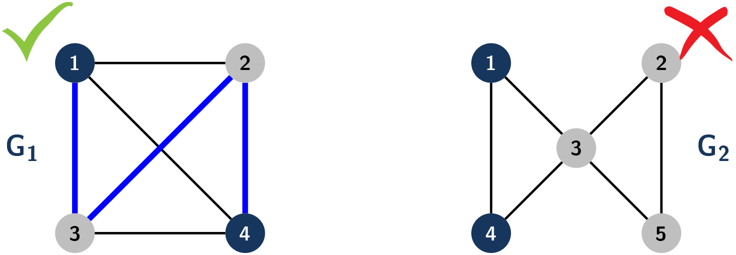
	Minimalna liczba ruchów
1	1
2	3
3	7
4	15
5	31
6	63
7	127
⋮	⋮
64	18 446 744 073 709 551 615
⋮	⋮
n	$\mathbf{2^n - 1}$

Nondeterministic Polynomial

NPTIME (NP)

$$NP = \bigcup_k NTIME(n^k)$$

$$HAMPATH = \left\{(G, v_1, v_2) : G \text{ zawiera ścieżkę Hamiltona } v_1 \rightsquigarrow v_2\right\}$$



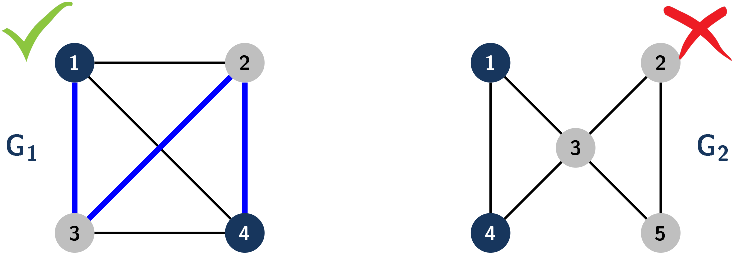
Klasa NPTIME

Nondeterministic Polynomial

NPTIME (NP)

NP jest klasą problemów posiadających wielomianowe algorytmy weryfikujące

$$HAMPATH = \Big\{(G, v_1, v_2) : G \text{ zawiera ścieżkę Hamiltona } v_1 \rightsquigarrow v_2\Big\}$$



P oraz NP

P Klasa problemów, które mogą być efektywnie rozwiązane (czas wielomianowy).

NP Klasa problemów, których rozwiązania mogą być efektywnie zweryfikowane (czas wielomianowy).

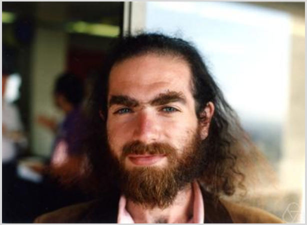
1. **P vs NP (1971)**
2. **Hipoteza Hodge'a (1950)**:
   Czy na algebraicznych rozmaitościach rzutowych każdy cykl Hodge'a jest wymierną liniową kombinacją cykli algebraicznych?
3. **Hipoteza Poincaré (1904)**
   Każda trójwymiarowa zwarta i jednospójna rozmaitość topologiczna bez brzegu jest homeomorficzna ze sferą trójwymiarową.
4. **Hipoteza Riemanna (1859)**
   Część rzeczywista każdego nietrywialnego zera funkcji dzeta jest równa $\frac{1}{2}$.
5. **Teoria Yanga-Millsa (1954)**
   Próba jednym formalizmem matematycznym oddziaływania słabego, silnego i elektromagnetycznego.
6. **Równania Naviera-Stokesa (1822)**
   Rozwiązania tych równań dla najbardziej skomplikowanych zjawisk hydrodynamicznych.
7. **Hipoteza Bircha i Swinnertona-Dyera (1960)**
   Związana z przewidywaniem rozwiązywalności każdego równania diofantycznego.

1. P vs NP **(1971)**

2. **Hipoteza Hodge'a (1950)**:
   Czy na algebraicznych rozmaitościach rzutowych każdy cykl Hodge'a jest wymierną liniową kombinacją cykli algebraicznych?

3. **Hipoteza Poincaré (1904) — (2003)**
   Każda trójwymiarowa zwarta i jednospójna rozmaitość topologiczna bez brzegu jest homeomorficzna ze sferą trójwymiarową.

Grigorij Perelman



4. **Hipoteza Riemanna (1859)**
   Część rzeczywista każdego nietrywialnego ze

5. **Teoria Yanga-Millsa (1954)**
   Próba jednym formalizmem matematycznym silnego i elektromagnetycznego.

6. **Równania Naviera-Stokesa (1822)**
   Rozwiązania tych równań dla najbardziej sko hydrodynamicznych.

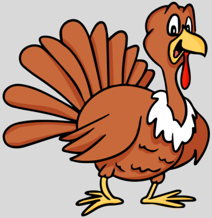
7. **Hipoteza Bircha i Swinnertona-Dyera (1960)**
   Związana z przewidywaniem rozwiązywalności każdego równania diofantycznego.

1. **P vs NP (1971)**

2. **Hipoteza Hodge'a (1950)**:
   Czy na algebraicznych rozmaitościach rzutowych każdy cykl Hodge'a jest wymi

   Donald Knuth

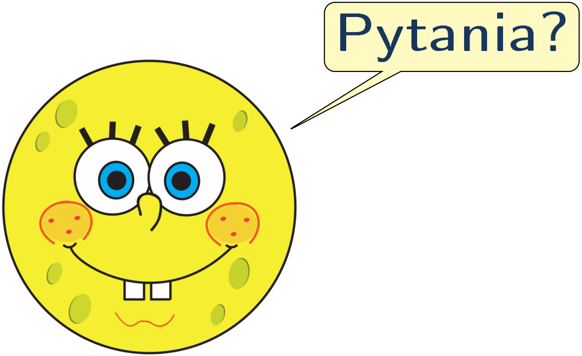
3. **Hipo**
   Każd topologiczna bez brzeg

4. **Hipo**
   Częś dzeta jest równa $\frac{1}{2}$.

5. **Teori**
   Próba ania słabego, silnego i elektromagnetycznego.

6. **Równania Naviera-Stokesa (1822)**
   Rozwiązania tych równań dla najbardziej skomplikowanych zjawisk hydrodynamicznych.

7. **Hipoteza Bircha i Swinnertona-Dyera (1960)**
   Związana z przewidywaniem rozwiązywalności każdego równania diofantycznego.


Pytania?