Zjemnělá složitost

Michal Koucký
Informatický ústav Univerzity Karlovy
MFF UK

Výpočetní složitost

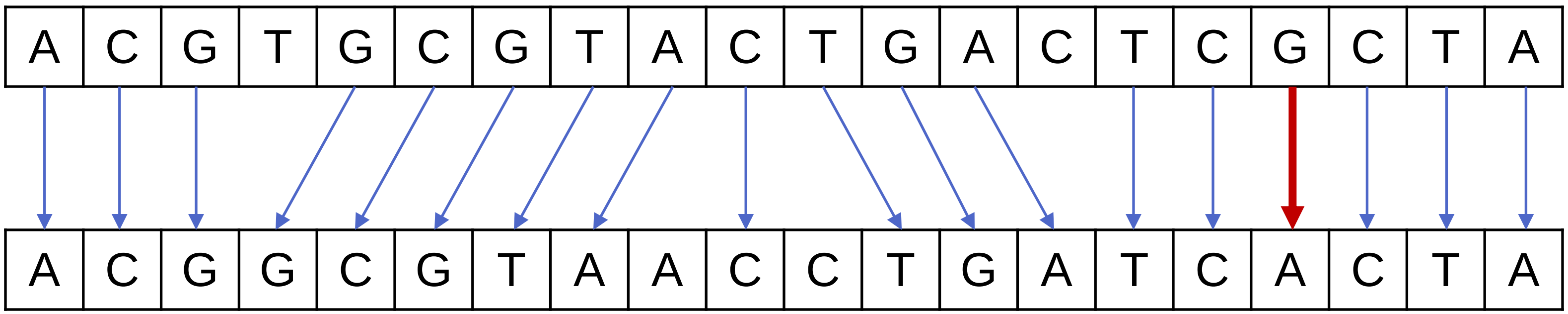
Klasifikace výpočetních problémů podle obtížnosti:

- *čas:* Kolik kroků výpočtu je třeba na vyřešení problému.
- *prostor:* Kolik paměti je třeba na vyřešení problému.

Další míry: množství *komunikace*, množství *náhodnosti*, *nedeterminismu*…

Podobnost řetězců

x



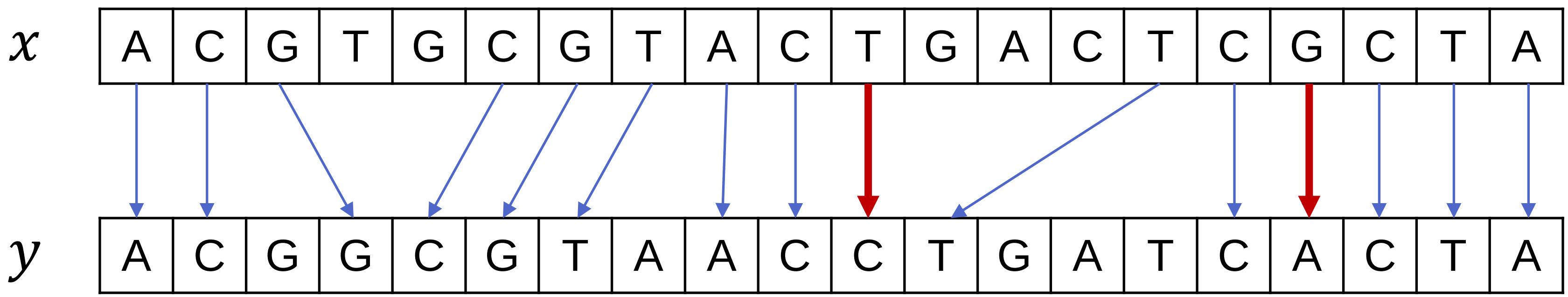
y

Editační vzdálenost x a y :

počet 1) změněných symbolů
2) vložených symbolů, a
3) vymazaných symbolů

které z x udělají y.

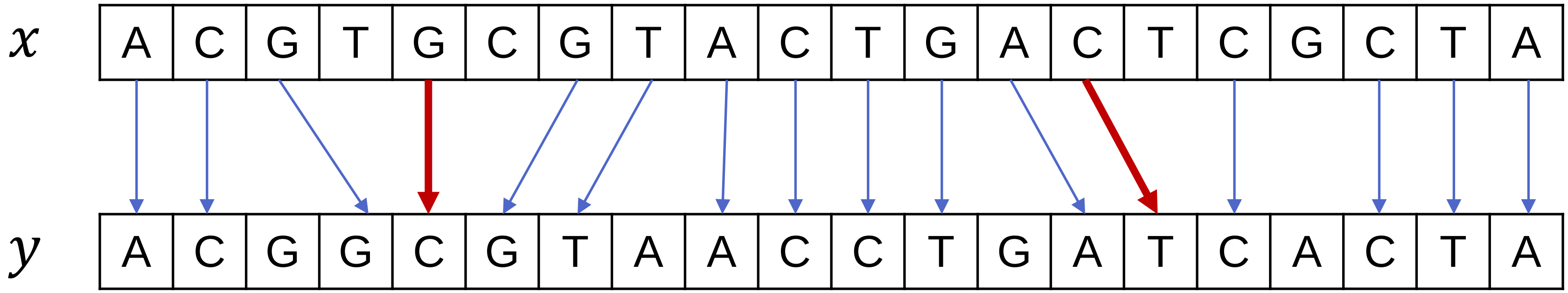
Podobnost řetězců



Zkouším všechny možnosti … 2^n kroků

n … délka x a y.

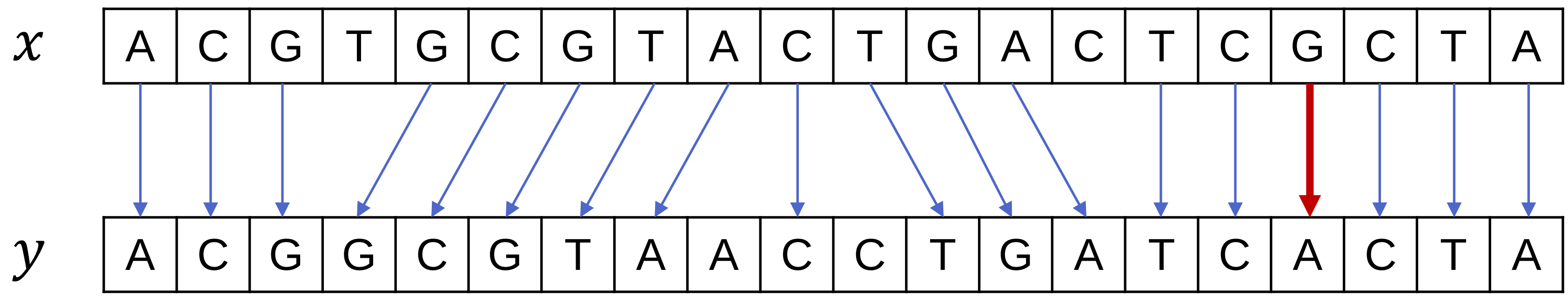
Podobnost řetězců



Zkouším všechny možnosti … 2^n kroků

n … délka x a y.

Podobnost řetězců

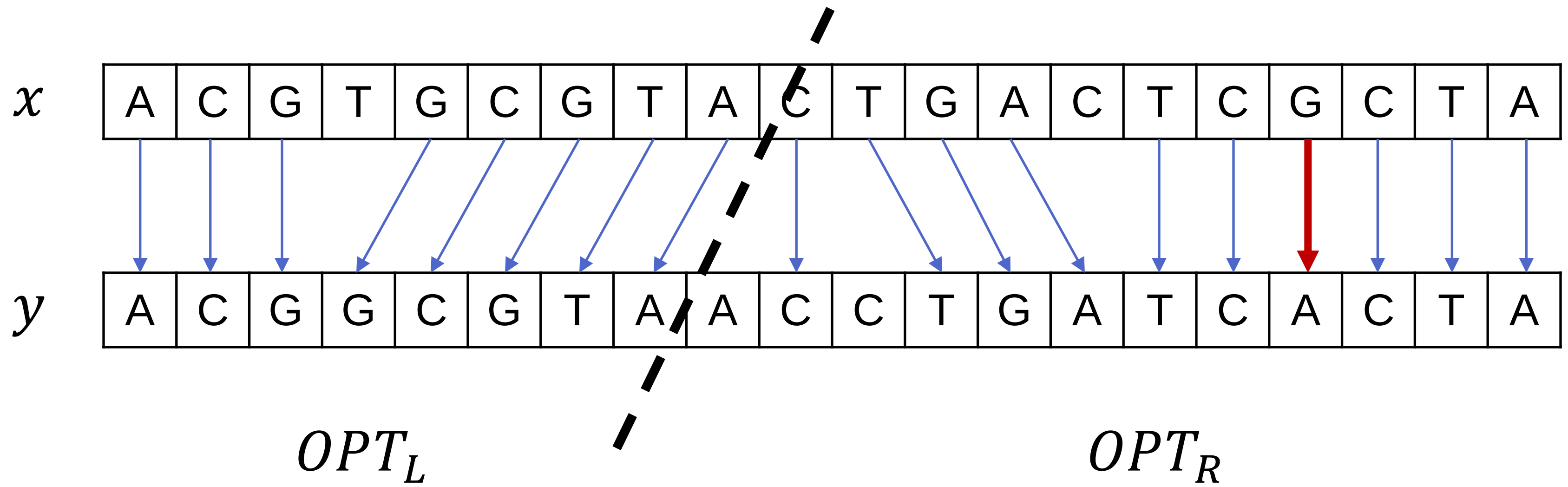


Zkouším všechny možnosti … 2^n kroků

exponenciální

n … délka x a y.

Podobnost řetězců

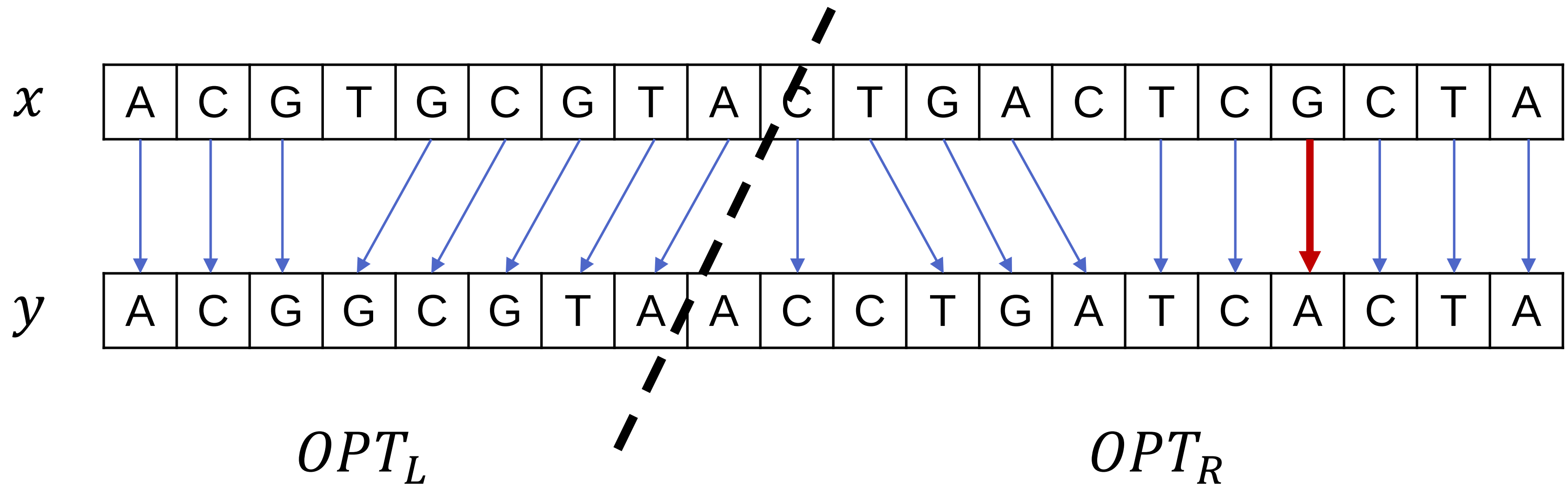


Podrozdělím na části … n^4 kroků

polynomiální

n … délka x a y.

Podobnost řetězců



Podrozdělím na části … $\cancel{n^4} \to n^2$ kroků

polynomiální

n … délka x a y.

Podobnost řetězců

nejlepší algoritmus $n^2/\log n$ (Masek-Paterson)

Aproximace

$$OPT \leq ALG \leq C \cdot OPT$$

pro nějakou konstantu C $n^{1.001}$ (Andoni-Nosatzki)

pro $C = 1.001$?

Heuristiky

rychlé, ale bez záruky

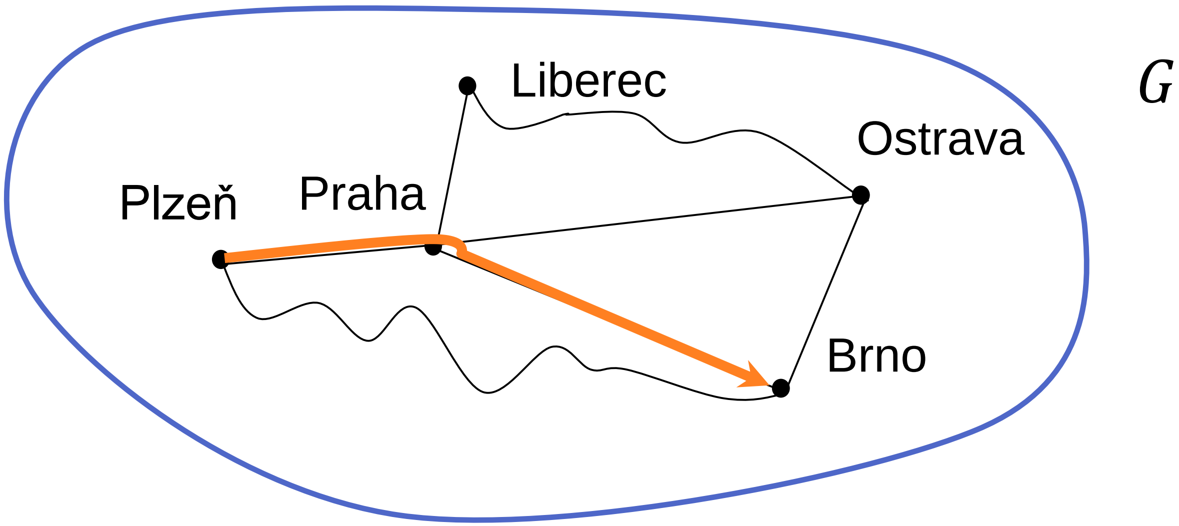
Čas běhu

n	n^2	n^3	n^4	2^n
10	100	1000	10^4	10^3
1 000	10^6	10^9	10^{12}	10^{300}
1 000 000	10^{12}	10^{18}	10^{24}	$10^{300\,000}$
1 000 000 000	10^{18}	10^{27}	10^{36}	$10^{300\,000\,000}$

superpočítač … 10^{18} operací/s
Xbox … 10^{13} operací/s

rok … $3 \cdot 10^7$ s
stáří vesmíru … $14 \cdot 10^9$ let

Nejkratší cesta v grafu



Úloha: Najdi nejkratší cestu z města A do města B.

Algoritmus (Dijkstra) … $m + n \log n$ kroků

n měst

m silnic

Tabulka vzdáleností v grafu (APSP)

	Brno	…	Liberec	Praha
Brno	0		307	206
…				
Liberec	308		0	113
Praha	207		110	0

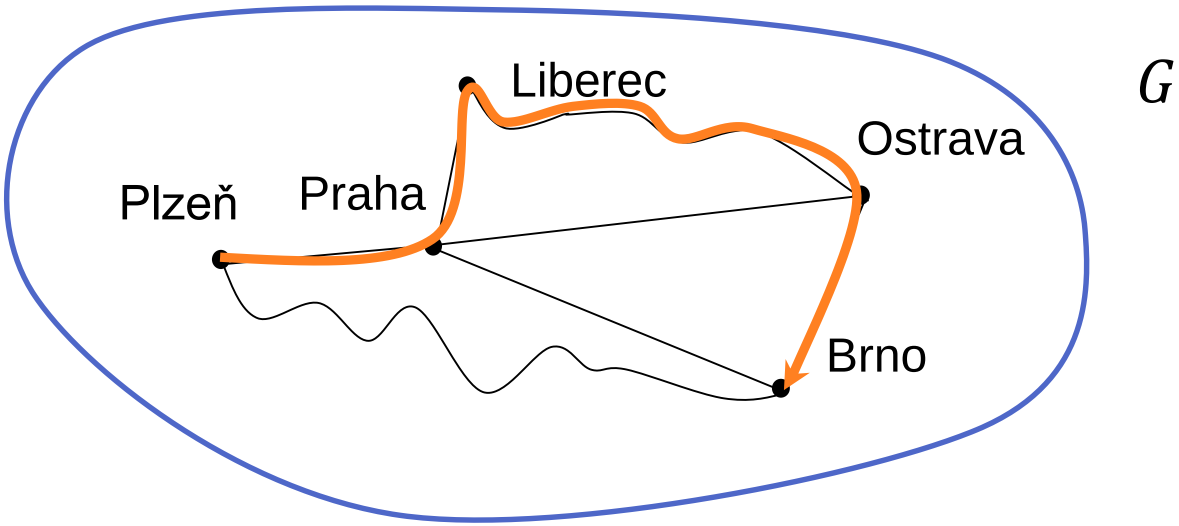
Úloha: Najdi nejkratší cestu pro každou dvojici měst.

Algoritmus (Floyd–Warshall/Dijkstra) … n^3 kroků

n měst

m silnic

Nejkratší *obchodní* cesta v grafu



Úloha: Najdi nejkratší cestu z města A do města B, která navštíví všechna města, každé ale jen jednou.

Algoritmus (Held-Karp) … $n^2 2^n$ kroků

n měst

m silnic

Splnitelnost formule

SAT: Je daná formule v CNF tvaru s n proměnnými a m klauzulemi splnitelná?

$$(x_1 \text{ OR } x_7) \text{ AND } (\neg x_3 \text{ OR } x_1 \text{ OR } x_5 \text{ OR } \neg x_7) \text{ AND } (x_3 \text{ OR } \neg x_5 \text{ OR } x_1)$$

Naivní algoritmus $\dots\ mn \cdot 2^n$

Nejlepší algoritmus $\dots\ mn \cdot 2^{(1-o(1))n}$

(Calabro-Impagliazzo-Paturi)

Silná hypotéza o exponenciálním čase

Silná hypotéza o exponenciálním čase (SETH):

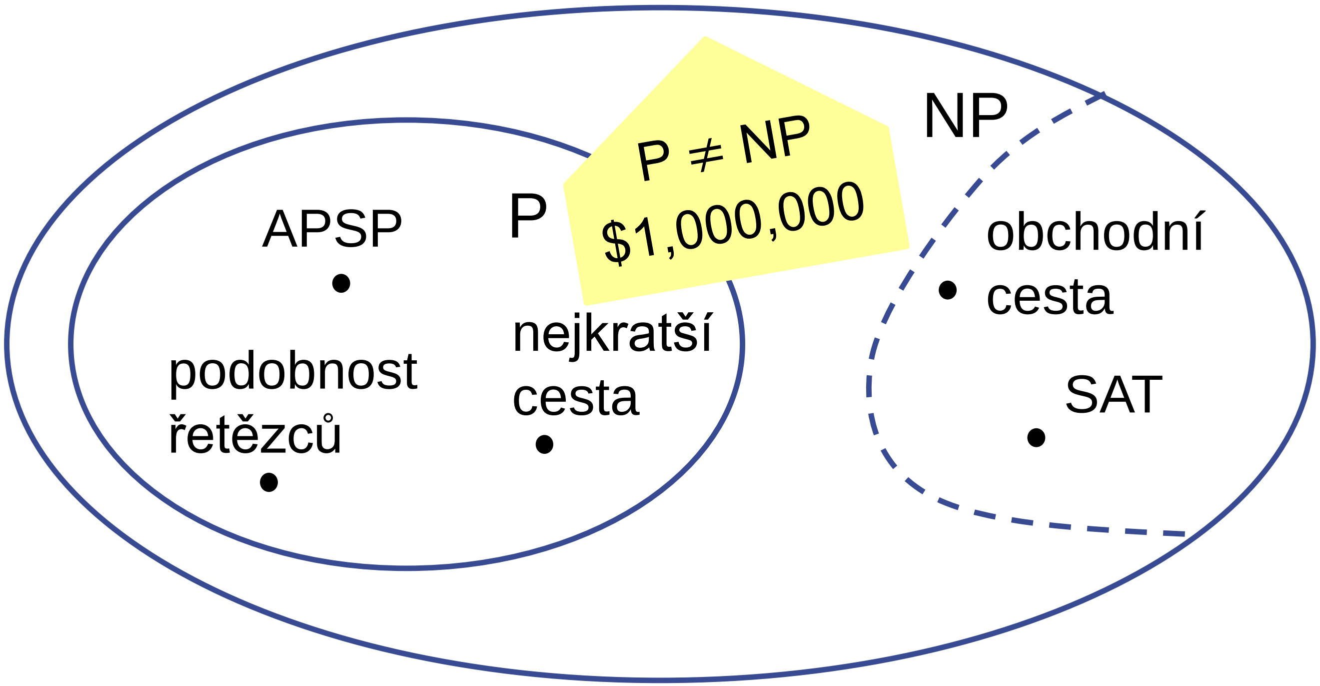
SAT nemá algoritmus pracující v čase $m \cdot 2^{(1-\epsilon)n}$ pro žádné pevné $\epsilon > 0$.

(Imagliazzo-Paturi-Zane)

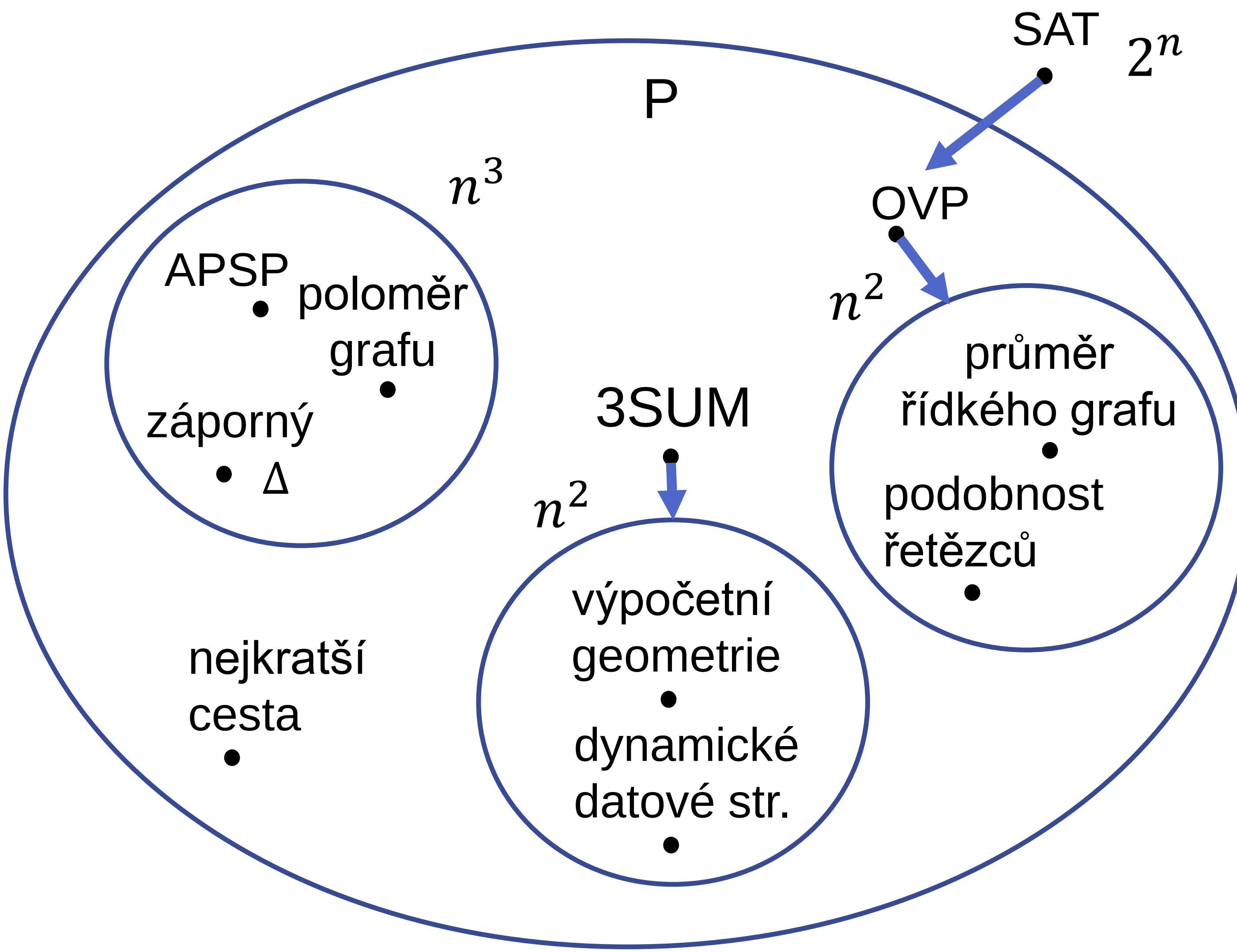
SAT: Je daná formule v CNF tvaru s n proměnnými a m klauzulemi splnitelná?

Klasická složitost

- P – umíme řešit v polynomiálním čase:
  podobnost řetězců, nejkratší cesta, APSP, …
- NP – řešení lze ověřit v polynomiálním čase:
  nejkratší obchodní cesta, splnitelnost (SAT)



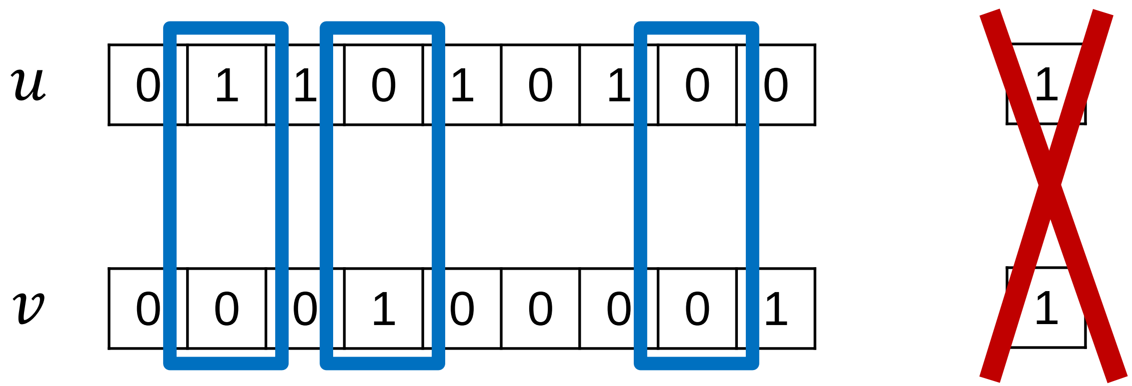
Zjemnělá složitost



Hledání kolmých vektorů (OVP)

Vstup: U, V množiny n vektorů z $\{0,1\}^d$

Cíl: Nalezni $u \in U$ a $v \in V$ takové, že $< u, v > = 0$.

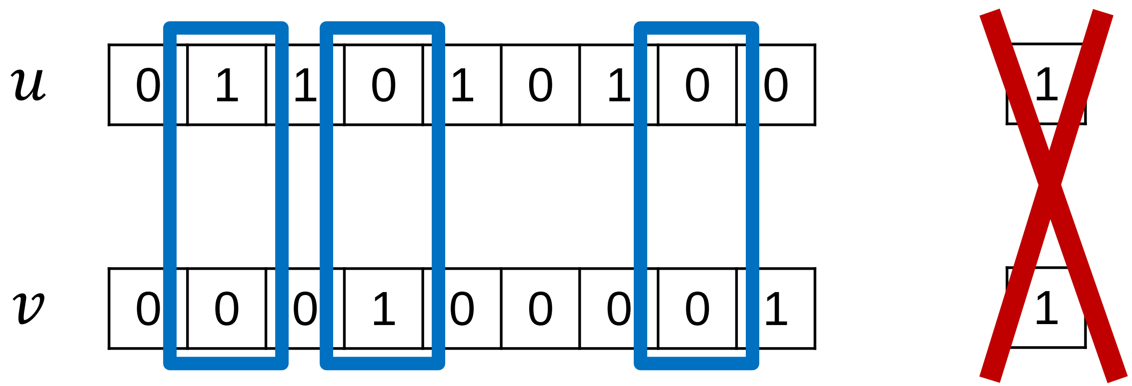


Naivní algoritmus … $n^2 d$ $d \ll n$

Hledání kolmých vektorů (OVP)

Vstup: U, V množiny n vektorů z $\{0,1\}^d$

Cíl: Nalezni $u \in U$ a $v \in V$ takové, že $< u, v > = 0$.



Tvrzení: Pokud OVP lze řešit v čase $n^{2-\varepsilon}$ pro pevné $\varepsilon > 0$, pak lze SAT řešit v čase $2^{(1-\varepsilon/2)n}$. (Williams)

Redukce SAT → OVP

Vstup: $\varphi(x_1, x_2, \ldots, x_n) = C_1 \text{ AND } C_2 \text{ AND} \cdots \text{AND } C_m$

Výstup: U, V množiny N vektorů z $\{0,1\}^d$

$$C_{10} = (\neg x_3 \text{ OR } x_1 \text{ OR } x_5 \text{ OR } \neg x_7)$$

$$x_1 = False, \quad x_2 = False, \quad x_3 = True, \quad x_4 = False,$$
$$x_5 = False, \quad x_6 = True, \quad x_7 = False, \quad x_8 = False$$

Redukce SAT → OVP

Vstup: $\cdots$ AND C_m

Výstup z $\{0,1\}^{m+2}$

i-tá pozice je 0
$\Longleftrightarrow$
C_i je splněno při ohodnocení
$x_1, \dots, x_{n/2}$ pomocí a

u_{FFF}

$_{FF}$

1	**0**	1	1	0	0	1

u_a

0	**1**	0	0	0	1	0

v_b

1	**0**	1	0	0	0	1

u_{TTT}

0	**1**	1	1	0	1	1

i-tá pozice je 0
$\Longleftrightarrow$
C_i je splněno při ohodnocení
$x_{1+n/2}, \dots, x_n$ pomocí b

$a \in \{F, T\}^{n/2}$

$< u_a, v_b > = 0 \quad \Longleftrightarrow \quad a \cdot$

Redukce SAT → OVP

Vstup: $\varphi(x_1, x_2, \dots, x_n) = C_1 \text{ AND } C_2 \text{ AND} \cdots \text{AND } C_m$

Výstup: U, V množiny $N = 2^{n/2}$ vektorů z $\{0,1\}^{m+2}$

u_{FFF}	**0**	**1**	0	1	0	1	1
u_a	**0**	**1**	0	0	0	1	0
u_{TTT}	**0**	**1**	1	1	0	1	1

$a \in \{F, T\}^{n/2}$

v_{FFF}	**1**	**0**	1	1	0	0	1
v_b	**1**	**0**	1	0	0	0	1
v_{TTT}	**1**	**0**	0	1	0	1	1

$b \in \{F, T\}^{n/2}$

$$N^{2-\varepsilon} = 2^{(1-\varepsilon/2)n}$$

Redukce

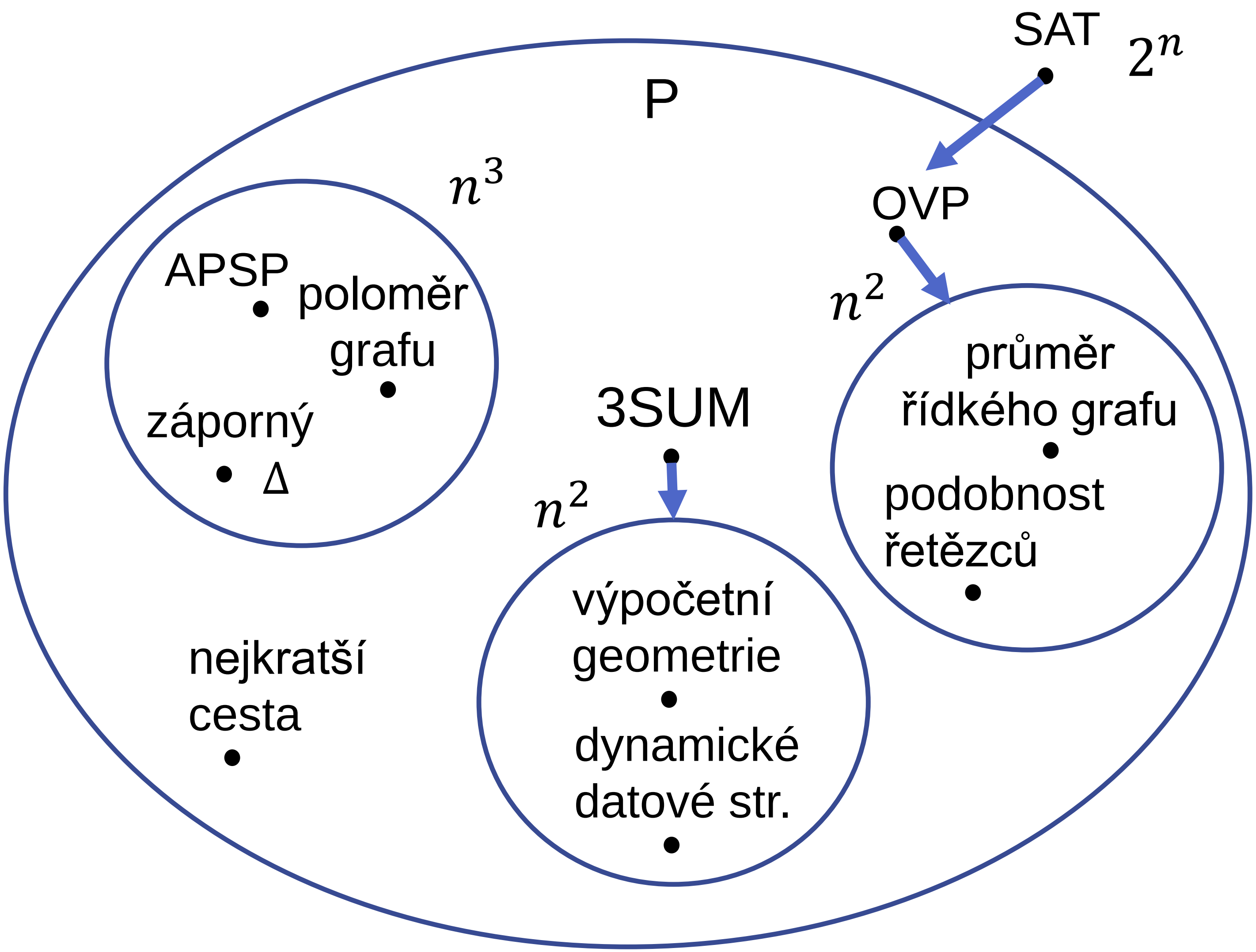
Tvrzení: Pokud podobnost řetězců lze řešit v čase $n^{2-\delta}$ pro pevné $\delta > 0$, pak OVP lze řešit v čase $n^{2-\varepsilon}$ pro $\varepsilon > 0$.

(Indyk-Backurs)

Tvrzení: Pokud OVP lze řešit v čase $n^{2-\varepsilon}$ pro pevné $\varepsilon > 0$, pak lze SAT řešit v čase $2^{(1-\varepsilon/2)n}$. (Williams)

podobnost řetězců → OVP → SAT

Zjemnělá složitost



KONEC