

Čas v počítači

Michal Koucký

Informatický ústav Univerzity Karlovy

MFF UK

Můj výzkum

Základní otázky týkající se výpočtů:

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

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

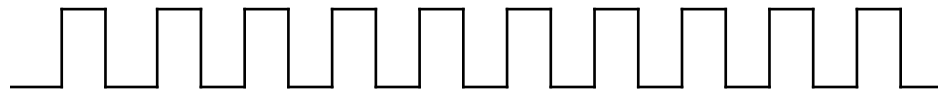
Čas v počítači

Chceme:

- rychlou odezvu
- výsledky co nejdříve

Čas v počítači

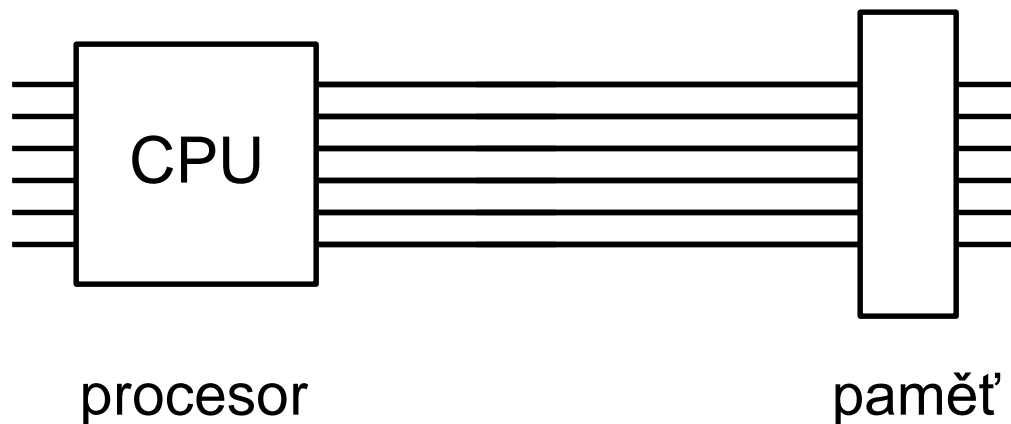
- Čas v počítači plyne v jednotlivých krocích.
- Řídí se hodinami:



takty

| | |
|------------|--------------|
| 1970. léta | 1 MHz |
| 1980. léta | 10-100 MHz |
| 1990. léta | 100-1000 MHz |
| 2000+ | 1-5 GHz |

Práce v počítači



Instrukce:

- sečti dvě čísla
- vynásob dvě čísla
- zapiš číslo do paměti
- ...

1970 1 instrukce = 5-30 taktů

2000 1 takt = 2 a více instrukcí

1970→2020 zrychlení 1 000 000x

Třídění - přímý výběr

14, 7, 8, 4, 20, 15, 21

n čísel



4, 7, 8, 14, 15, 20, 21

Třídění - přímý výběr

14, 7, 8, ~~4~~ 20, 15, 21

↑ ↑ ↑ ↑ ↑ ↑ ↑

n čísel

min=~~4~~4

4

Třídění - přímý výběr

14, ~~1~~, 8, ~~2~~ 20, 15, 21

↑ ↑ ↑ ↑ ↑ ↑ ↑

n čísel

min=~~1~~4

4, 7

Třídění - přímý výběr

~~x~~, ~~x~~, ~~x~~, ~~x~~, ~~x~~, ~~x~~, ~~x~~
↑ ↑ ↑ ↑ ↑ ↑ ↑

n čísel

min=...

4, 7, 8, 14, 15, 20, 21

Třídění - přímý výběr

~~4~~, ~~7~~, ~~8~~, ~~14~~, ~~15~~, ~~20~~, ~~21~~

n čísel

min=...

4, 7, 8, 14, 15, 20, 21

→ $n \times n$ operací

Třídění - QuickSort

14, 7, 8, 4, 20, 15, 21

n čísel

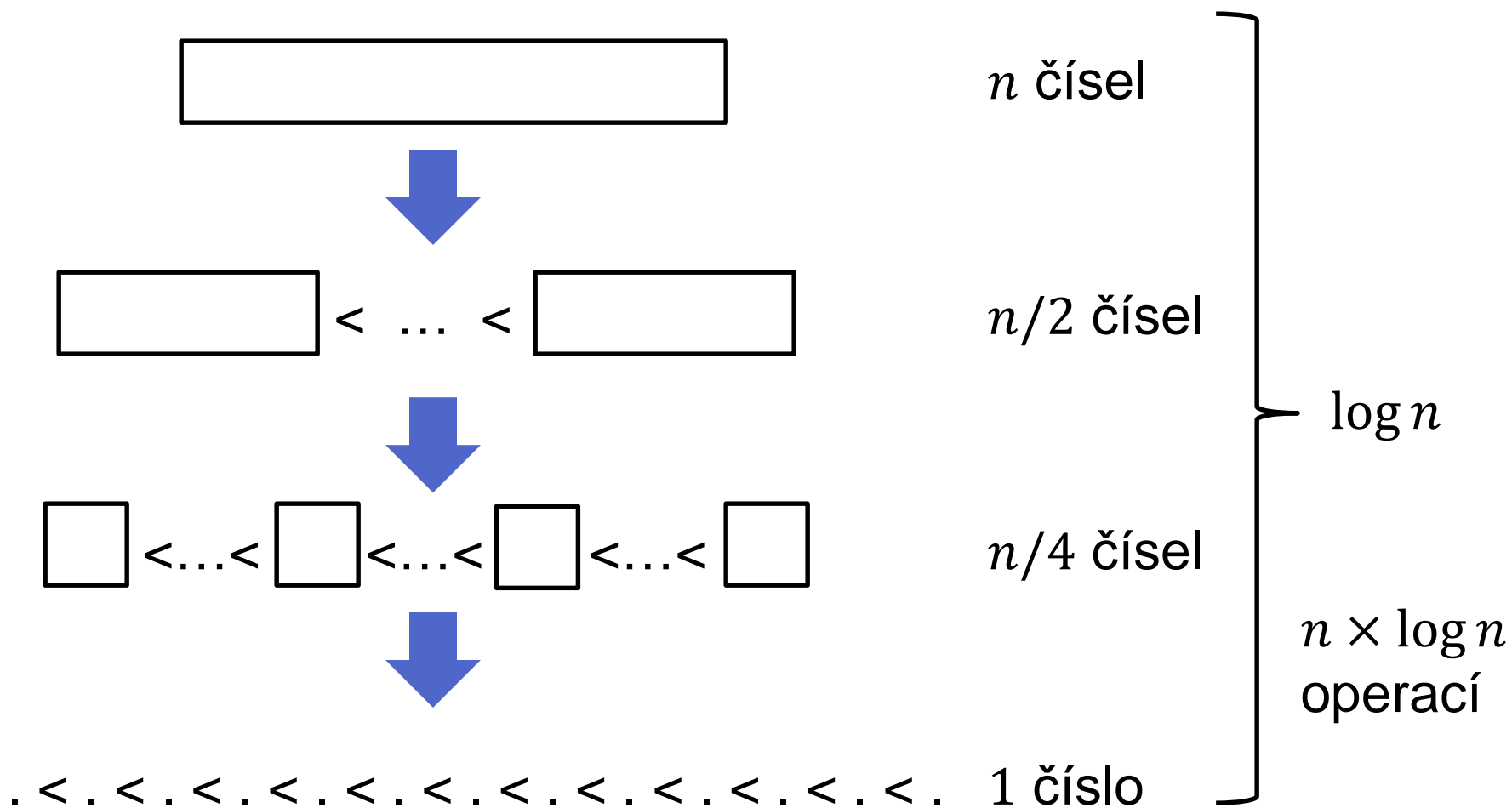


7, 8, 4 < 14 < 20, 15, 21

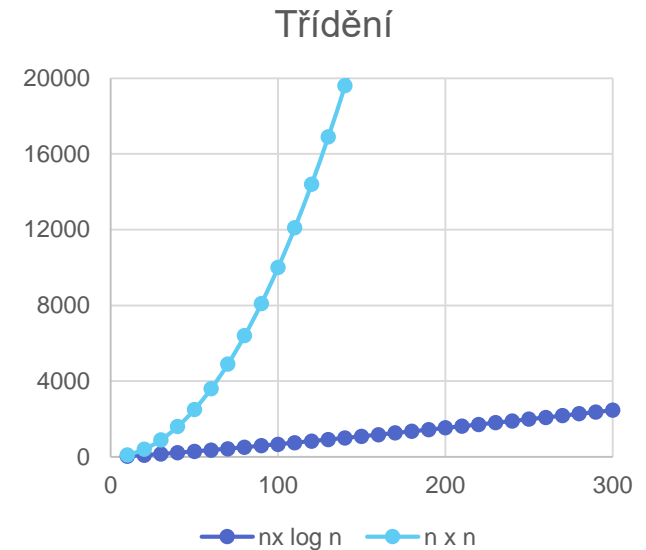


4 < 7 < 8 < 14 < 15 < 20 < 21

Třídění - QuickSort



Třídění - počet operací



| n | přímý výběr n^2 |
|-----------|----------------------|
| 10 | 100 |
| 100 | 10 000 |
| 1 000 | 1 000 000 |
| 1 000 000 | 1 000 000 000 000 |

QuickSort

$n \times \log n$

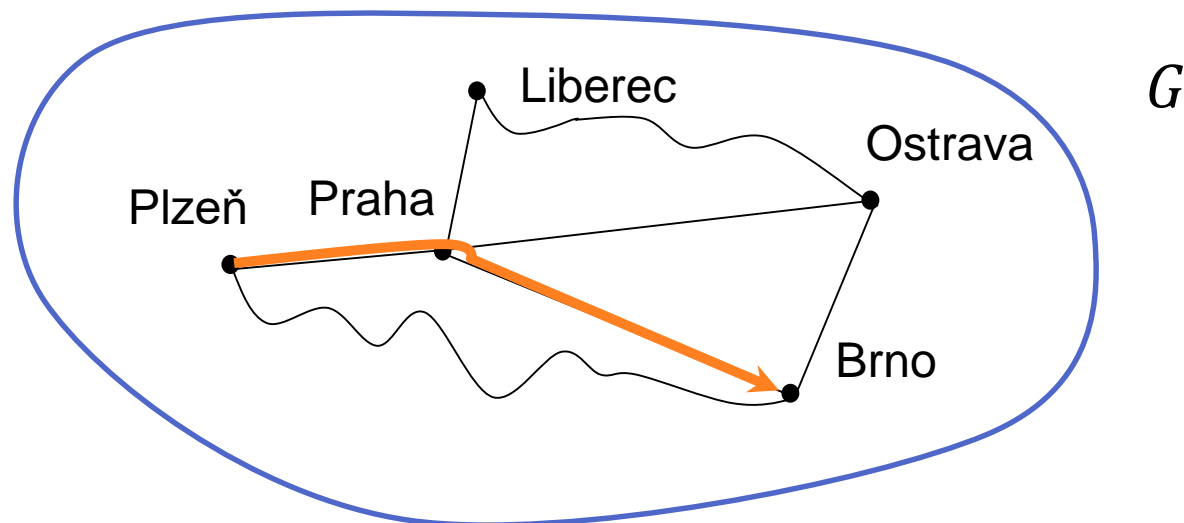
30

700

10 000

20 000 000

Nejkratší cesta v grafu



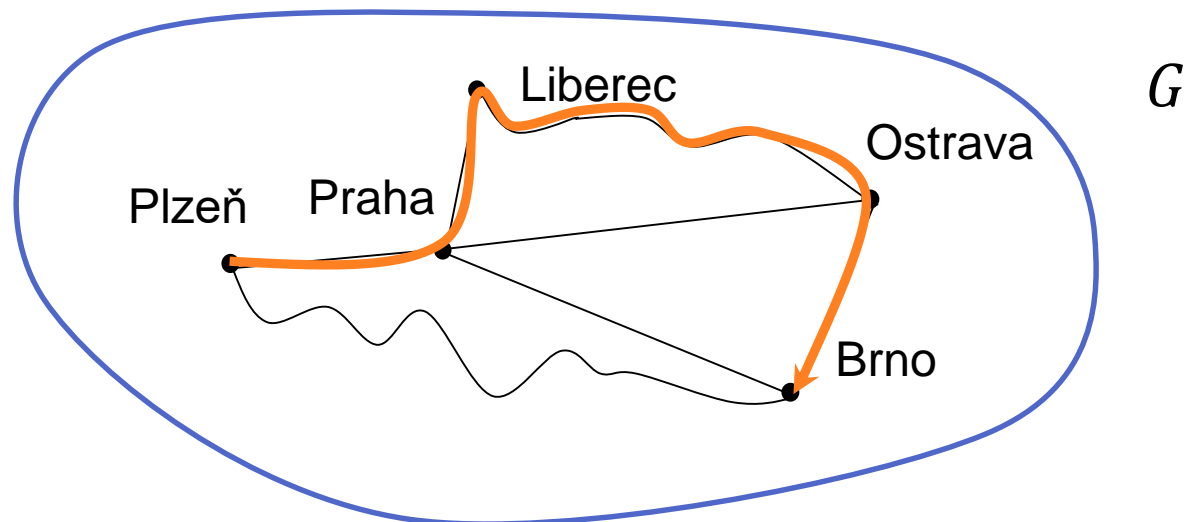
Úloha: Najdi nejkratší cestu z města A do města B .

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

n měst

m silnic

Nejkratší *obchodní* cesta v grafu



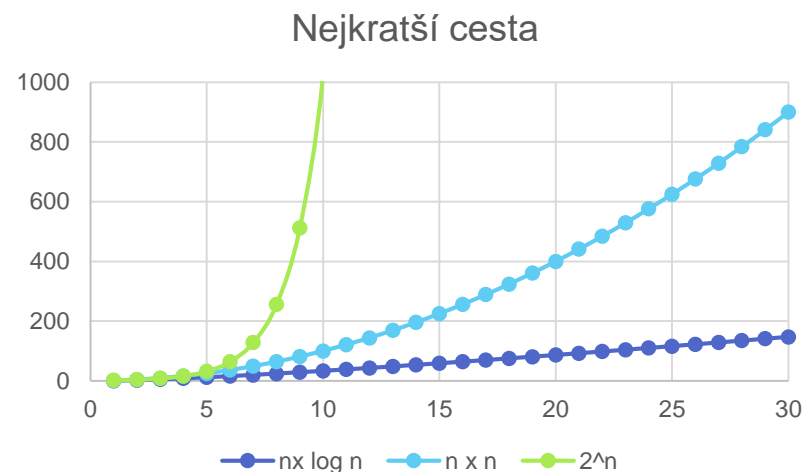
Úloha: Najdi nejkratší cestu z města A do města B , která navštíví všechna města, každé ale jen jednou.

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

n měst

m silnic

Nejkratší cesta



n nejkratší cesta

$n \log n$

10 30

20 90

30 150

40 200

obchodní cesta

2^n

1 000

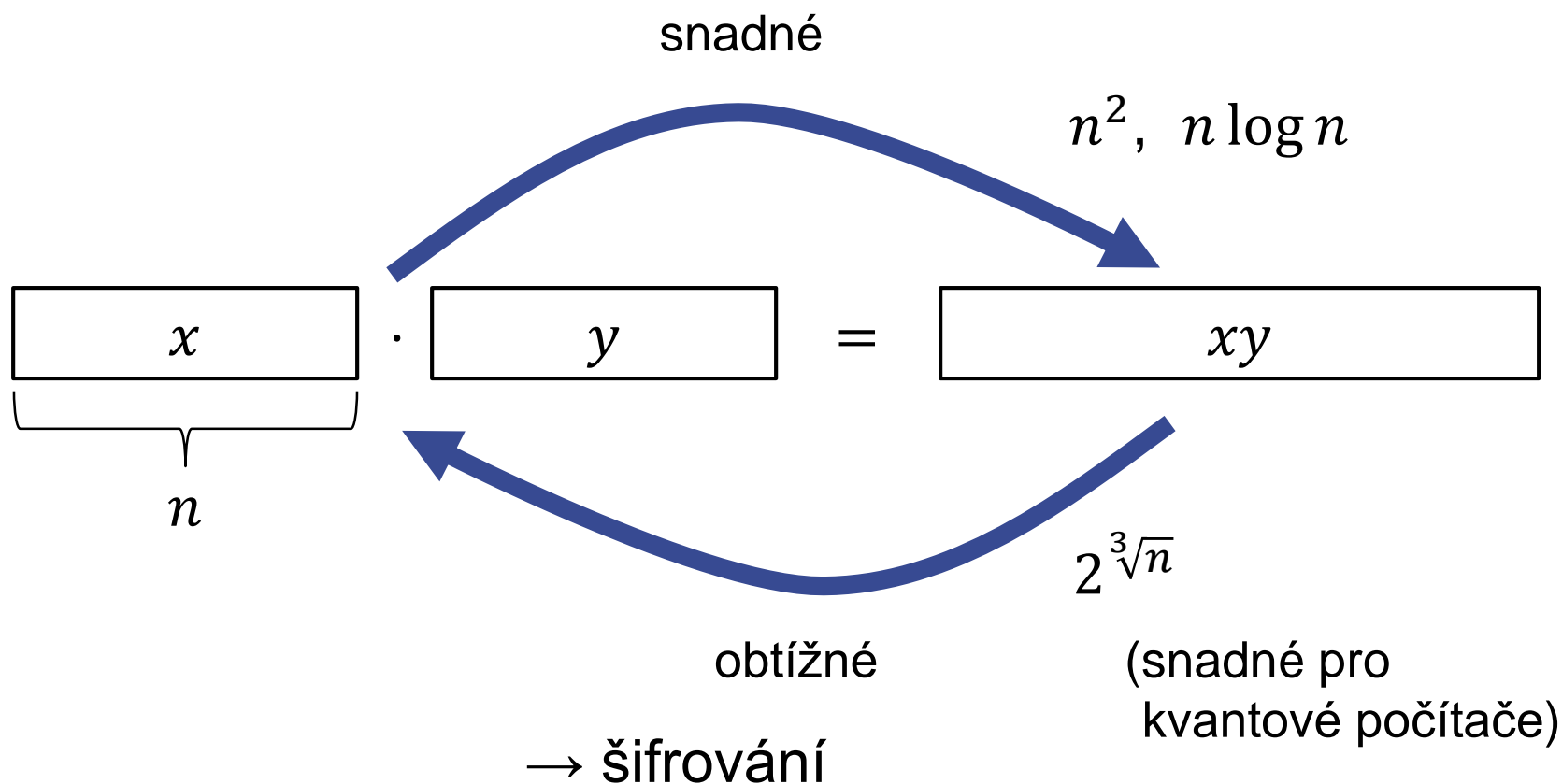
1 000 000

1 000 000 000

1 000 000 000 000

Rychlý, nemusí být dobrý

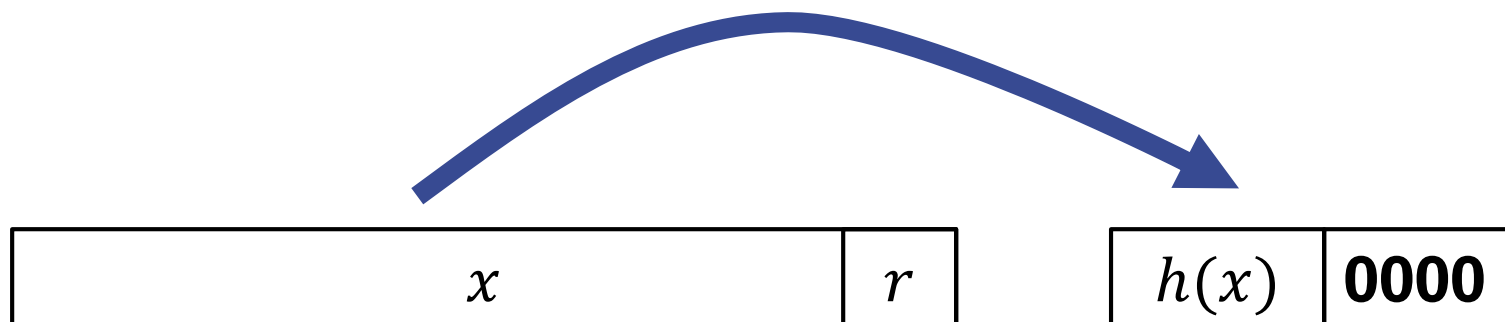
Chceme co nejrychlejší algoritmy, ale ne vždy:



Bitcoin

Těžba bitcoinu:

$h...$ hašovací funkce, snadné



najdi vhodné r , obtížné

Problém zastavení programu

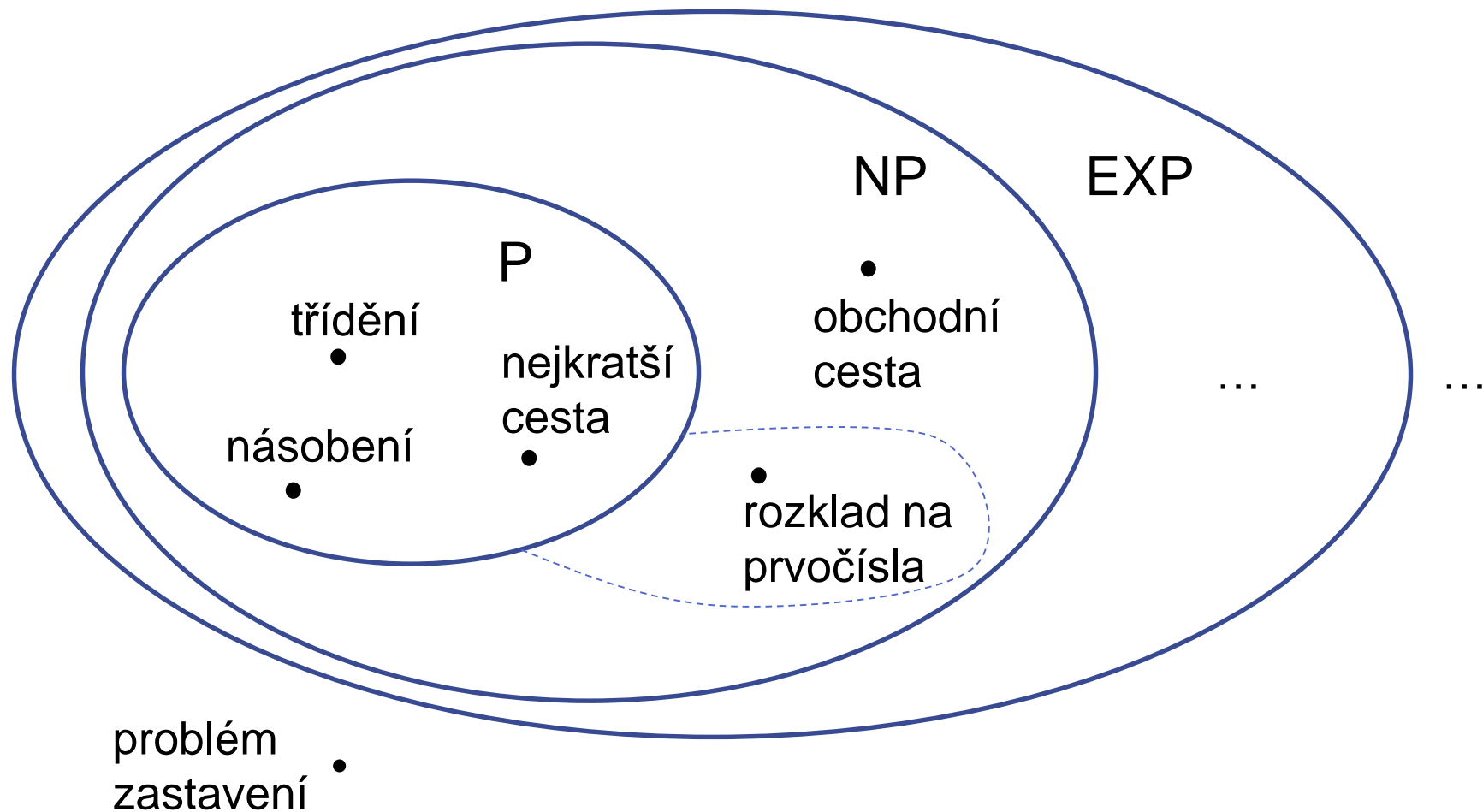
```
i=100  
while i>0 do  
    i = i+1
```

```
i=100  
while i>0 do  
    i = i-1
```

Turing 1936: Nelze rozhodovat žádným algoritmem, který program se zastaví a který ne.

... *Halting problem*

Výpočetní složitost



KONEC