# Složitost

# Složitost algoritmu

instance → algoritmus → řešení

čas
paměť

Jak dlouho bude trvat řešení této instance?

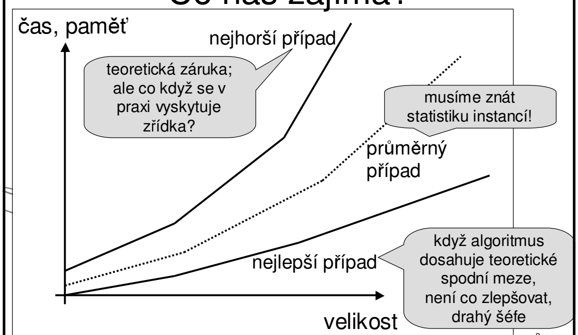
Obecně stejně těžké jako úlohu vyřešit; lépe vybrat si nějakou charakteristiku instance

velikost instance

složitost je funkce z velikosti do času/paměti



# Co nás zajímá?






# Měření složitosti

- Jak měřit velikost instance?
  - hrubá míra: počet prvků instance (uzlů, čísel, prvků množiny)
  - jemná míra: počet bitů, nutných k zakódování instance
- Jak měřit čas výpočtu?
  - počet „typických operací"
  - počet kroků jednotného výpočetního modelu

jaké zakódování?

co to je?

Turingův stroj na obzoru



# Typické operace

- Příklady operací (při řešení typickým algoritmem)
  - Hledání
    - srovnání
  - Řazení
    - porovnání a výměna
  - Násobení matic
    - násobení dvou čísel
  - Problém batohu a jiné kombinatorické problémy
    - Výpočet omezení a optimalizačního kritéria konfigurace



# Asymptotická složitost

Nechť $f, g: \mathbf{N} \rightarrow \mathbf{R}^+$

Asymptotická horní mez

$f(n) = O(g(n)) \Leftrightarrow \exists c > 0,\ n_0 \in \mathbf{N}: \forall n > n_0 : f(n) \leq c.g(n)$

Asymptotická dolní mez

$f(n) = \Omega(g(n)) \Leftrightarrow \exists c > 0,\ n_0 \in \mathbf{N}: \forall n > n_0 : f(n) \geq c.g(n)$

Asymptotický odhad

$f(n) = \Theta(g(n)) \Leftrightarrow f(n) = O(g(n))\ \&\ f(n) = \Omega(g(n))$

# Únosné a neúnosné výpočty

- Pro dané *n*, algoritmus složitosti *O*(g(*n*)) skončí v:

| g(*n*) / *n* | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| ***n*** | 1 ms | 2 ms | 3 ms | 4 ms | 5 ms |
| ***n*.log *n*** | 10 ms | 26 ms | 44 ms | 64 ms | 85 ms |
| $n^2$ | 100 ms | 400 ms | 900 ms | 1.6 s | 2.5 s |
| $2^n$ | 1 s | 17 min | 12 d | 25 r | $4 \cdot 10^4$ r |
| ***n*!** | 1 h | $8 \cdot 10^7$ r | $8 \cdot 10^{21}$ r | $3 \cdot 10^{37}$ r | $1 \cdot 10^{54}$ r |
| $n^n$ | 115 d | $3 \cdot 10^{15}$ r | $7 \cdot 10^{33}$ r | $4 \cdot 10^{53}$ r | $3 \cdot 10^{74}$ r |



# Únosné a neúnosné výpočty

- Algoritmus složitosti *O*(g(*n*)) provede následující množství výpočtů, jestliže pro n=10 trvá výpočet 1 min:

| g(*n*) | 1 min pro *n* = | 1 hodina pro *n* = | 1 den pro *n* = | 1 rok pro *n* = |
|---|---|---|---|---|
| ***n*** | 10 | 600 | 14 400 | 5 256 000 |
| ***n*.log *n*** | 10 | 250 | 3 997 | 853 895 |
| $n^2$ | 10 | 77 | 379 | 7 249 |
| $2^n$ | 10 | 15 | 20 | 29 |
| ***n*!** | 10 | 11 | 12 | 15 |
| $n^n$ | 10 | 11 | 12 | 14 |



# Únosné a neúnosné výpočty

- Algoritmus složitosti *O*(g(*n*)) provede následující množství výpočtů na ideálním paralelním stroji daného rozměru, jestliže pro n=10 trvá výpočet 1 min na 1 procesoru:

| g(*n*) | 1 CPU pro *n* = | 600 CPU pro *n* = | 14 400 CPU pro *n* = | 5 256 000 CPU pro *n* = |
|---|---|---|---|---|
| ***n*** | 10 | 600 | 14 400 | 5 256 000 |
| ***n*.log *n*** | 10 | 250 | 3 997 | 853 895 |
| $n^2$ | 10 | 77 | 379 | 7 249 |
| $2^n$ | 10 | 15 | 20 | 29 |
| ***n*!** | 10 | 11 | 12 | 15 |
| $n^n$ | 10 | 11 | 12 | 14 |



# Složitost v praxi

Jeden problém, dva algoritmy a dvě platformy

| Jazyk | C | Java |
|---|---|---|
| **Algoritmus** | **Bubble sort** | **Quick sort** |
| ***n* = 4** | **1 ms** | **10 ms** |
| ***n* = 8** | **64 ms** | **240 ms** |
| ***n* = 16** | **256 ms** | **690 ms** |
| ***n* = 32** | **1024 ms** | **1600 ms** |
| ***n* = 64** | **4096 ms** | **3840 ms** |
| ***n* = 128** | **16384 ms** | **8960 ms** |

v praxi, jsme zde...

vždy najdeme průsečík

nebo zde...?



# Složitost problému

- Umíme měřit složitost algoritmu. Jak přejít ke složitosti problému?

Problém $\Pi$ má složitost $O(g(n))$, jestliže pro $\Pi$ existuje algoritmus, který má složitost $O(g(n))$

Problém $\Pi$ má složitost $\Omega(g(n))$, jestliže každý algoritmus řeší $\Pi$ in $\Omega(g(n))$

každý, i ten, který neznáme (důkazy...)



# Asymptotické meze

## Asymptotická složitost

Význam?

$f(n) = O(g(n)) \Leftrightarrow \exists c > 0,\ n_0 \in \mathbf{N}: \forall n > n_0 : f(n) \leq c.g(n)$

- $n_0$ – malé instance nejsou důležité (asymptoticky…)
- $c$ – konstanty nejsou důležité



## Asymptotický odhad

Vlastnosti:

Nechť

$f_1(n) = \Theta(g_1(n))$

$f_2(n) = \Theta(g_2(n))$

Pak

$f(n) = f_1(n) + f_2(n) \Rightarrow f(n) = \Theta(\max(g_1(n), g_2(n)))$

$f(n) = f_1(n)\ .\ f_2(n) \Rightarrow f(n) = \Theta(g_1(n)\ .\ g_2(n))$



## Asymptotický odhad

Další vlastnosti

- reflexivní
- symetrický
- tranzitivní

Platí také pro $O(g(n))$ a $\Omega(g(n))$.



## Asymptotický odhad

- Příklady

Nechť $f(n) = \Theta(n^2)$

Pak

$f(n) = O(n^2)$, $f(n) = \Omega(n^2)$

$f(n) = \Theta(2n^2)$

$f(n) = O(n^3)$, $f(n) = O(2^n)$, $f(n) = O(n!)$

$f(n) = \Omega(n)$, $f(n) = \Omega(1)$