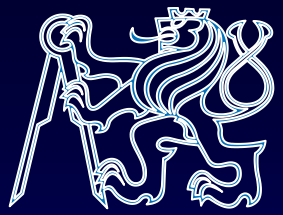
- ©Jan Schmidt 2011
  Katedra číslicového návrhu
  Fakulta informačních technologií
  České vysoké učení technické v Praze

- Zimní semestr 2013/14



EVROPSKÝ SOCIÁLNÍ FOND
PRAHA & EU: INVESTUJENE
DO VAŠÍ BUDOUCNOSTI

MI-PAA

1. Kombinatorické problémy a algoritmy

- Kombinatorický problém
- Problém, instance, řešení instance
- Vstupní, výstupní a konfigurační proměnné
- Omezující podmínky a optimalizační kritérium
- Rozhodovací, konstruktivní a optimalizační verze
- Výpočetní složitost algoritmu → výpočetní složitost problému
- Únosné a neúnosné problémy

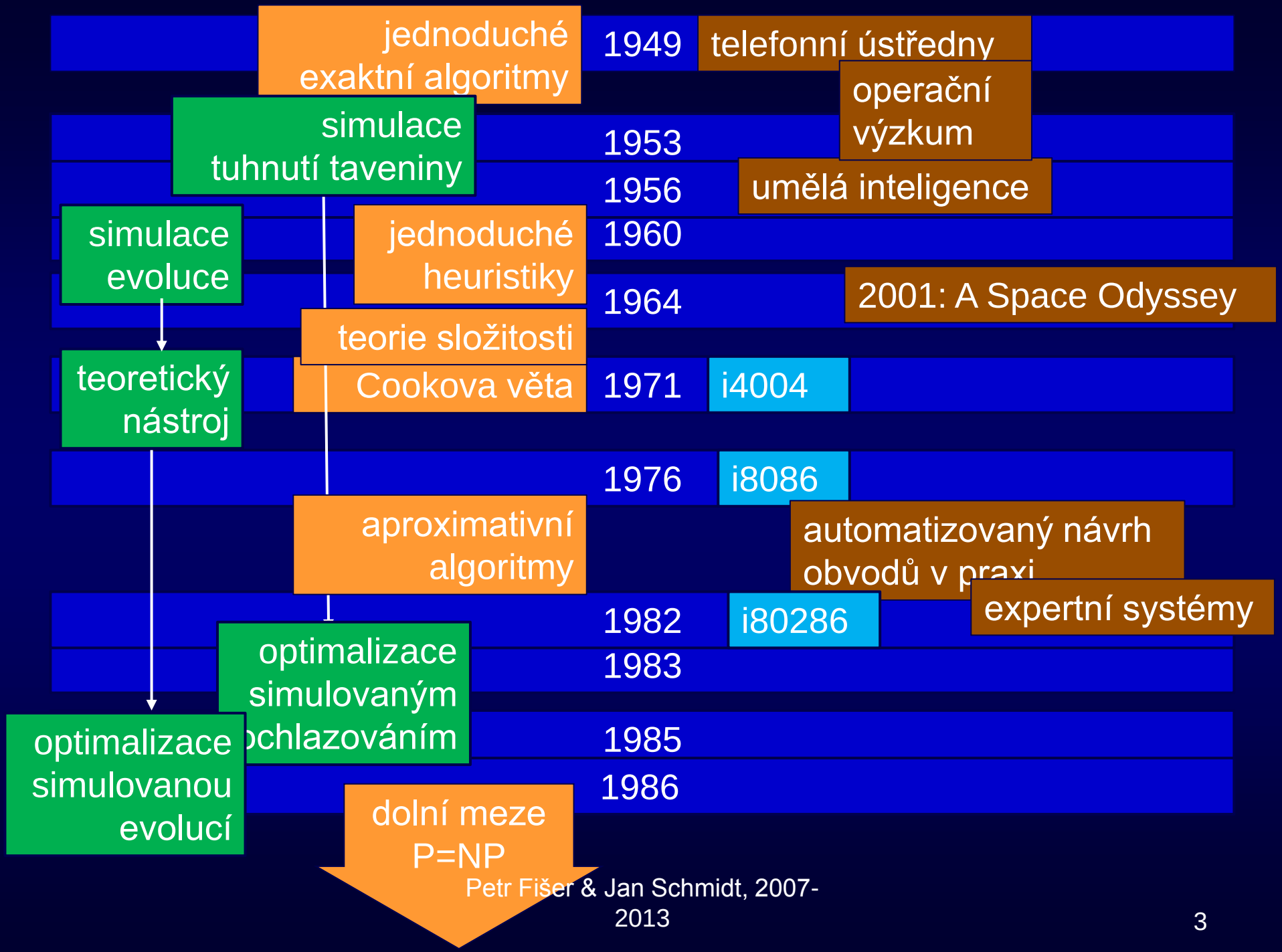
KOMBINATORICKÁ MATEMATIKA JE … EHM … JE KDYŽ SE ZKOUŠEJÍ RŮZNÉ KOMBINACE

… a když to funguje

Petr Fišer & Jan Schmidt, 2007-2013

Kombinatorická matematika

- zajímá se o konečné a diskrétní problémy
- konečný počet proměnných
- konečný počet hodnot pro každou proměnnou
- tudíž hrubá síla vždy funguje:
  - vyzkoušet všechny kombinace hodnot všech proměnných
  - poskytne výsledek v konečném čase, tudíž
  - je to algoritmus
- … ale většinou není prakticky použitelná


jednoduché exaktní algoritmy
1949
telefonní ústředny
operační výzkum
simulace tuhnutí taveniny
1953
1956
umělá inteligence
simulace evoluce
jednoduché heuristiky
1960
1964
2001: A Space Odyssey
teorie složitosti
teoretický nástroj
Cookova věta
1971
i4004
1976
i8086
aproximativní algoritmy
automatizovaný návrh obvodů v praxi
expertní systémy
1982
i80286
optimalizace simulovaným ochlazováním
1983
1985
optimalizace simulovanou evolucí
1986
dolní meze P=NP

- problém
- instance
- konfigurace
- vstupní, výstupní proměnné
- konfigurační proměnné
- omezující podmínky
- optimalizační kritérium
- řešení, optimální řešení, suboptimální řešení

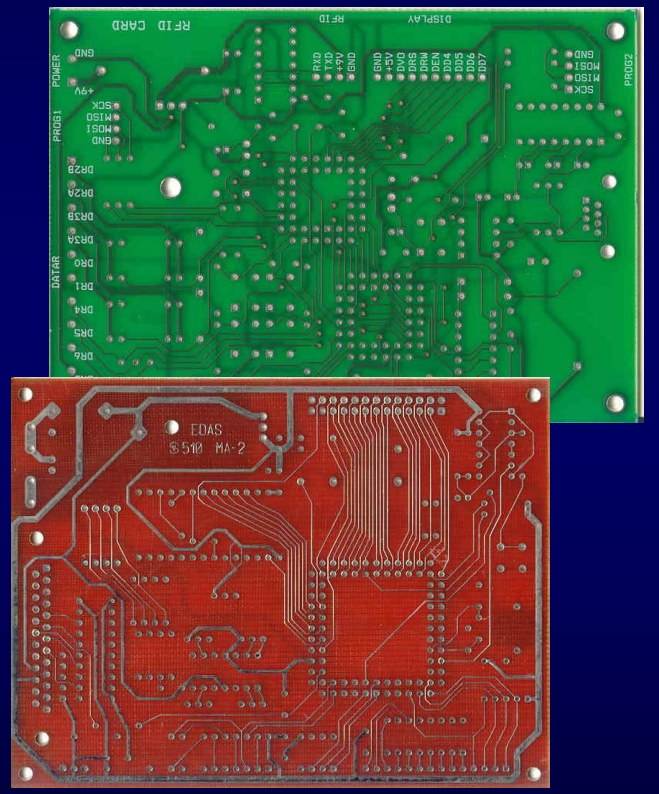
TERMINOLOGIE KOMBINATORICKÝCH PROBLÉMŮ

Problém a instance

problém

Nalézt optimální (nejkratší) cestu pro NC vrtačku na dané desce, která začíná a končí v předepsané klidové pozici.

instance



Kombinatorický problém

Charakterizován:

- vstupními proměnnými
- výstupními proměnnými
- konfiguračními proměnnými
- omezením
- optimalizačním kritériem, pokud je třeba

Konfigurační proměnné – to, co nastavuje hrubá síla Proměnných

- je konečný počet
- mají konečné domény

Příklad

vstupní proměnné
výstupní proměnné
konfigurační proměnné
omezení
optimalizační kritérium

seznam děr

pořadí děr

pořadí děr

- uzavřená cesta
- každá díra právě jednou

nejkratší

Instance a řešení problému

instance problému	ohodnocení vstupních proměnných
konfigurace	ohodnocení konfiguračních proměnných
řešení instance	konfigurace, která splňuje omezení
optimální řešení	řešení s nejlepší hodnotou optimalizačního kritéria
suboptimální řešení	řešení s vyhovující hodnotou optimalizačního kritéria

při řešení nějaké instance

čemu?

Problém batohu

Jsou dána přirozená čísla n, M, c_1, c_2, c_3, …, c_n, w_1, w_2, w_3, …, w_n. Nalezněte čísla x_1, x_2, x_3, …, x_n z množiny {0,1} tak, aby

$$\sum_{i=1}^{n} x_i w_i \leq M \qquad \sum_{i=1}^{n} x_i c_i = \max.$$

Je dáno n věcí, i-tá věc má váhu w_i a cenu c_i, dále batoh s nosností M. Nalezněte takovou sestavu věcí v batohu, aby nebyl přetížen a cena věcí byla maximální.

Problém batohu

Vstupní proměnné: n, M, $W = \{w_1, w_2, \ldots, w_n\}$, $C = \{c_1, c_2, \ldots, c_n\}$

Konfigurační proměnné: $X = \{x_1, x_2, \ldots, x_n\}$, $x_i \in \{0, 1\}$

Výstupní proměnné: tytéž

Omezení:

$$\sum_{i=1}^{n} x_i w_i \leq M$$

Optimalizační kritérium:

$$\sum_{i=1}^{n} x_i c_i = \max.$$

Instance a konfigurace

instance $n=3$, $M=6$, $C=\{10, 20, 30\}$, $W=\{2, 3, 5\}$

všechny konfigurace

X_1	X_2	X_3	$\Sigma x_i c_i$	$\Sigma x_i w_i$		
0	0	0	0	0	✓	triviální
0	0	1	30	5	✓✔	
0	1	0	20	3	✓	
0	1	1	50	8	⊘	
1	0	0	10	2	✓	
1	0	1	40	7	⊘	
1	1	0	30	5	✓✔	
1	1	1	60	10	⊘	

✓ řešení

✔ optimální

Verze kombinatorických problémů

Nechť I je instance, Y konfigurace, $R(I, Y)$ omezení (tj. $R(I, Y)$ říká, zda Y je řešením)

- rozhodovací problém
  Existuje Y takové, že $R(I, Y)$?
  Pro všechna Y, platí, že $R(I, Y)$?
- konstruktivní problém
  Sestrojit nějaké Y takové, že $R(I, Y)$.
- enumerační problém
  Sestrojit všechna Y taková, že $R(I, Y)$.

Všechny tyto problémy mají společné
- vstupní proměnné
- konfigurační proměnné
- omezení R
- jenom výstup je jiný

Různé verze téhož problému

Verze kombinatorických problémů

Všechny tyto problémy mají společné
- vstupní proměnné
- konfigurační proměnné
- omezení *R*
- jenom výstup je jiný

- Podle toho je rozpoznám: hledám frázi „Existuje“ „Platí pro všechny“ „Sestrojit (nalézt, zkonstruovat nějaké (všechny)“ …
- Podle toho je převádím: ponechám vstupní a konfigurační proměnné, změním výstup

inženýrské úlohy vždy něco optimalizují !?

Optimalizační problémy

Nechť *I* je instance, *Y* konfigurace, *R*(*I*, *Y*) omezení a *C(Y)* optimalizační kritérium (cenová funkce)

- optimalizační rozhodovací problém

  Existuje *Y* takové, že *R*(*I*, *Y*) a *C(Y)* je aspoň tak dobré jako daná konstanta *Q*?
- optimalizační konstruktivní problém

  Sestrojit nějaké *Y* takové, že *R*(*I*, *Y*) a *C(Y)* je nejlepší možné.
- optimalizační enumerační problém

  Sestrojit všechna *Y* taková, že *R*(*I*, *Y*) a *C(Y)* je nejlepší možné.
- optimalizační evaluační problém

  Zjistit nejlepší možné *C(Y)* takové, že *R*(*I*, *Y*).

Rozhodovací problémy a jazyky

- Vstupní proměnné zakóduji pomocí množiny symbolů, nejjednodušeji binárně
- Každá instance je tedy charakterizována řetězem 0 a 1 (prvkem {0,1}*)
- Problém charakterizuji všemi instancemi s výstupem ANO
- Je to tedy podmnožina {0,1}*
- Je to tedy jazyk
- Rozhodovací verze problému batohu:

- Definice problému splnitelnosti
- Rozhodovací, konstruktivní a enumerativní verze
- Optimalizační verze
- Různé známé varianty optimalizační verze

VZTAH VERZÍ PROBLÉMU NA PŘÍKLADU PROBLÉMU SPLNITELNOSTI BOOLEOVSKÉ FORMULE

Problém splnitelnosti booleovské formule (SAT)

splnitelnost = satisfiability
→ SAT

Dáno:

n proměnných $X = (x_1, x_2, \ldots x_n)$

Booleovská formule $F(X)$ v konjunktivní normální formě (součin součtů)

Příklad: $F(X) = (x_1 + x_2' + x_3)\ (x_1' + x_2)\ (x_1' + x_3')$

Nalézt:

Je tato formule splnitelná?

Tj. existuje ohodnocení $Y = (y_1, y_2, \ldots y_n)$ proměnných $x_1, x_2, \ldots x_n$ takové, že $F(Y) = 1$?

Výstup příkladu: ano

Konfigurace příkladu *(svědek)*: $y_1 = 1,\ y_2 = 1,\ x_3 = 0$

Verze SAT

Rozhodovací verze

- Existuje ohodnocení Y takové, že $F(Y)=1$?

Konstruktivní verze

- Sestrojit ohodnocení Y takové, že $F(Y)=1$

Enumerační verze

- Sestrojit všechna ohodnocení Y taková, že $F(Y)=1$

Optimalizační SAT

Optimalizační rozhodovací verze

- Existuje ohodnocení Y takové, že $F(Y)=1$ a Y má méně než Q jedniček?

Optimalizační konstruktivní verze

- Sestrojit ohodnocení Y takové, že $F(Y)=1$ a Y má co nejméně jedniček.

Optimalizační enumerační verze

- Sestrojit všechna ohodnocení Y taková, že $F(Y)=1$ a Y má co nejméně jedniček.

Verze SAT - shrnutí

	rozhod.	konstr.	enum.	opt. rozhod.	opt. konstr.	opt. enum.
vstup	F, X	F, X	F, X	F, X	F, X	F, X
konfigu-race	Y	Y	Y	Y	Y	Y
výstup	ano-ne	Y	$\{Y\}$	ano-ne	Y	$\{Y\}$
omezení	$F(Y)=1$	$F(Y)=1$	$F(Y)=1$	$F(Y)=1$	$F(Y)=1$	$F(Y)=1$
opt. kritérium	---	---	---	max. počet jedniček	max. počet jedniček	max. počet jedniček

SAT „folklór“

		MAX WEIGHTED SAT	MAX SAT	
vstup	F, X	F, X, W	F, X	F, X, W
konfigurace	Y	Y	Y	Y
výstup	Y	Y	Y	Y
omezení	$F(Y) = 1$	$F(Y) = 1$	---	---
opt. kritérium	max. počet jedniček	max. vážený počet jedniček	max. počet splněných termů	max. vážený počet splněných termů

Vztah verzí problémů

- Inženýrská praxe – často optimalizační problémy
- Teoretická odvození – rozhodovací problémy (jednoduchý výstup)
- Mají na to matematici právo?
- Optimalizační problém neúnosně složitý $\Leftrightarrow$ rozhodovací verze neúnosně složitá
- Proč …
- Co je to neúnosně …
- Dá se v těch složitostech udělat nějaký pořádek?

$\rightarrow$ teorie složitosti

Inženýrská optimalizace …

u toho většinou zůstaneme

- Jednoduchý případ:
  - mám celé zadání (instance) najednou – např. plánování výjezdů na příští den
  - vím co chci (umím to vyjádřit optimalizačním kritériem)
- Nemám celé zadání
  - např. plánování výjezdů podle přicházejících požadavků - on-line problémy
- Nevím pořádně co chci
  - mám více optimalizačních kritérií, a chci si vybírat různé možnosti - multikriteriální optimalizace
  - chci pokud možno různá suboptimální řešení - multimodální optimalizace

- Složitost algoritmu jako jeho abstrakce
- Nejhorší, nejlepší a statisticky očekávaná složitost
- Měření velikosti instance
- Měření výpočetní složitosti
- Únosné a neúnosné problémy
- Složitost problému a třídy složitosti

SLOŽITOST ALGORITMU A PROBLÉMU

Složitost algoritmu

Výpočetní prostředky: čas, paměť, …

Jak dlouho bude trvat řešení této instance?

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

složitost je funkce z velikosti do času/paměti

velikost instance

Co nás zajímá?

Asymptotická složitost

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

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

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

Měření složitosti: velikost instance

- Jak měřit velikost instance?
  - hrubá míra: počet prvků instance (uzlů, čísel, prvků množiny)
  - jemná míra: počet bitů, nutných k zakódování instance

jaké zakódování?

Měření složitosti: prostředky

BI-ZDM 0

BI-GRA 12

- Jak měřit potřebu prostředků?
  - počet „typických operací“?
  - počet paměťových míst?
- Jak dokázat vlastnosti algoritmu?
  - potřebujeme formálně definovaný výpočetní model
  - jaké prostředky nás zajímají, takový volíme model
  - počet kroků, komunikace…

Turingův stroj a jiné přístroje 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

Výpočetní modely

BI-GRA 12

- Turingův stroj, RAM stroj
  - paměťové médium + řízení stavovým strojem
  - algoritmizovatelnost (rozhodnutelnost) úloh
  - čas a paměť sekvenčního výpočtu
- Booleův obvod
  - síť hradel (logických operátorů)
  - prostředky: počet hradel
  - čas: hloubka sítě
- Komunikující bloky
  - každý dostane polovinu dat, kolik bitů si musí vyměnit?

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 =	60 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 µs	10 µs
$n = 8$	64 µs	240 µs
$n = 16$	256 µs	690 µs
$n = 32$	1024 µs	1600 µs
$n = 64$	4096 µs	3840 µs
$n = 128$	16384 µs	8960 µs

v praxi, jsme zde...

vždy najdeme průsečík

nebo zde...?

Složitost v praxi

v praxi, jsme zde...

$g(n)$	1 min pro n =	1 hodina pro n =	1 den pro n =
n	10	600	14 400
$n.\log n$	10	250	3 997
n^2	10	77	379
2^n	10	15	20
$n!$	10	11	12
n^n	10	11	12

nebo zde...?

Typickou doménou kombinatorické matematiky jsou složité problémy a velké instance

Složitost problému

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

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

Problém Π má složitost $\Omega(g(n))$, jestliže každý algoritmus řeší Π v čase (s pamětí) $\Omega(g(n))$

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

Třídy problémů: princip

- Dán výpočetní model
- Zvolíme omezení:
  - počet kroků polynomiální s velikostí instance
  - paměť polynomiální s velikostí instance
  - konstantní hloubka obvodu
  - logaritmický počet komunikovaných bitů
- Co se s tímto omezeným prostředkem dá spočítat?
  - $\rightarrow$ třída problémů
- Jaké problémy jsou v té třídě nejtěžší?
  - srovnání obtížnosti $\rightarrow$
    $\rightarrow$ vzájemný převod (redukce) instancí

Čemu teď rozumíme

Čím je dán problém a čím instance

Jak pro daný problém najdu konfigurační proměnné

Čím se liší rozhodovací, konstruktivní a jiné verze problému a jak je odvodím jednu z druhé

Jak je charakterizována složitost problému (na rozdíl od složitosti algoritmu)

V čem spočívá neúnosnost takto označovaných problémů

Jaké pojmy k tomu potřebujeme

Kombinatorický problém

Problém, instance, řešení instance

Vstupní, výstupní a konfigurační proměnné

Omezující podmínky a optimalizační kritérium

Rozhodovací, konstruktivní a optimalizační verze

Výpočetní složitost algoritmu $\rightarrow$ výpočetní složitost problému

Únosné a neúnosné problémy