# Randomizované algoritmy

## Algoritmus pro MAX *k* SAT

- Booleovská formule *F* v konjunktivní formě proměnných *X*, *k* literálů v každé klauzuli, nalézt ohodnocení *Y* proměnných *X* tak, aby bylo splněno co nejvíce klauzulí.
- Algoritmus: každou proměnnou ohodnotíme 0 nebo 1 se stejnou pravděpodobností.
- Vlastnosti:
  pro splnitelné formule o *c* klauzulích, kde má každá klauzule alespoň *k* literálů, je očekávaná hodnota optimalizačního kritéria

$$\left(1-\frac{1}{2^k}\right).c$$

maximum je *c*

## Důkaz

- Pravděpodobnost, že klauzule o *k* literálech není splněna ---------- $2^{-k}$
- Pravděpodobnost, že klauzule o *k* literálech je splněna ---------- $1-2^{-k}$
- Očekávaný počet splněných klauzulí ---------- $c.(1-2^{-k})$

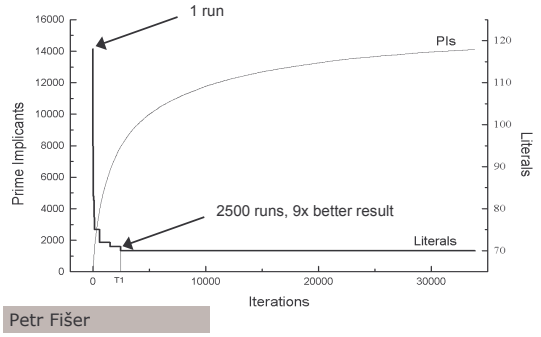
Q.E.D.

$$\left(1-\frac{1}{2^k}\right).c$$

## Miller-Rabinův test prvočíselnosti

- Dáno číslo *n*, zjistit, zda je prvočíslem.
- Malá věta Fermatova:
  Jestliže *n* je prvočíslo a *k* přirozené číslo, $1 \leq k < n$, pak $k^{n-1} \equiv 1 \pmod{n}$.
- Jestliže pro čísla *n*, *k* tvrzení malá věta Fermatovy neplatí, *k* nazveme svědkem složenosti čísla *n*.
- Jestliže *n* je složené číslo, pak ¾ přirozených čísel menších než *n* jsou svědky složenosti *n*.
- Otestujeme-li 100 náhodných potenciálních svědků, pak pravděpodobnost, že o *n* neprávem tvrdíme, že je prvočíslem, je $¼^{100}$.
- Tuto pravděpodobnost můžeme tedy libovolně snižovat.

## Minimalizace booleovských výrazů




Petr Fišer

## Randomizovaný algoritmus

- Založen na náhodné volbě
- Jeho vlastnosti jsou vyjádřeny statisticky
  - dosažený výsledek (optimalizační kritérium) je náhodná proměnná, čas běhu pevný pro danou instanci → Monte Carlo algoritmy
  - čas běhu je náhodná proměnná, výsledek vždy správný → Las Vegas algoritmy

*Monte Carlo: z herny se vypotácíte ráno, ale nevíte, kolik peněz vám zůstane*
*Las Vegas: vždycky vás oberou na kost, otázka je do kdy*

## Quicksort

- Výsledek vždy správně (seřazený)
- Čas běhu záleží na volbě pivotu
- Pivot musí být alespoň přibližně správně zvolen
- Typický Las Vegas algoritmus

Hra „Quicksort je k ničemu"

Hráč č. 1 předloží implementaci quicksortu.
Hráč č. 2 se snaží strefit do instance, pro kterou implementace volí pivoty špatně a čas je $O(n^2)$.

Když se to podaří do $k$ pokusů, vyhrává č. 2

Pokud implementace má randomizovanou volbu pivotu, č. 2 může vyhrát jen s velmi malou pravděpodobností

## Analýza randomizovaných algoritmů

- Poskytuje očekávanou (střední) hodnotu charakteristické veličiny (kvality, času)
- Platí pro jakýkoli vstup
- Pravděpodobnostní analýza poskytuje průměrné hodnoty při předpokládaném rozložení charakteristik vstupních instancí

## Randomizovaný 3 SAT

Booleovská formule $F$ v konjunktivní formě proměnných $X$, 3 literály v každé klauzuli, nalézt ohodnocení $Y$ proměnných $X$ tak, aby $F(Y)=1$.

- Algoritmus:
  1. Počáteční ohodnocení $Y$: každou proměnnou ohodnoť 0 nebo 1 se stejnou pravděpodobností.
  2. Pokud existuje ohodnocení $Y'$, které se liší od $Y$ v právě jedné proměnné, pak $Y \leftarrow Y'$ a opakuj 2
- Vlastnosti:
  pro každé $0<\varepsilon<½$ se dá vyjádřit pravděpodobnost, že algoritmus nalezne řešení, jako funkce $n$ a $\varepsilon$. Výrok pak platí pro všechny splnitelné formule až na jistou část, jejíž velikost je opět funkcí $n$ a $\varepsilon$.

## Randomizovaný 3 SAT

- Kombinace randomizované a deterministické fáze
- Z hlediska heuristických algoritmů, kombinace
  - náhodné konstruktivní fáze
  - deterministické iterativní fáze
- Praktičtější podoba, obecný SAT:
  - algoritmus GSAT
  - zaujatá náhodná procházka

## GSAT
(Selman, Levesque, and Mitchell 1992)

- Algoritmus:
  1. Počáteční ohodnocení $Y$: každou proměnnou ohodnoť 0 nebo 1 se stejnou pravděpodobností.
  2. Najdi ohodnocení $Y'$, které se liší od $Y$ v právě jedné proměnné a poskytne nejvíce splněných klauzulí
  3. $Y \leftarrow Y'$. Pokud nejsou všechny klauzule splněny nebo vyčerpán stanovaný počet kroků, opakuj 2.
  4. Pokud stále nejsou všechny klauzule splněny, opakuj s jiným náhodným počátečním ohodnocením.

## Náhodná procházka
(Random Walk)

- Algoritmus:
  1. Počáteční ohodnocení $Y$: každou proměnnou ohodnoť 0 nebo 1 se stejnou pravděpodobností.
  2. Najdi ohodnocení $Y'$, které se liší od $Y$ v právě jedné proměnné některé nesplněné klauzule
  3. $Y \leftarrow Y'$. Pokud nejsou všechny klauzule splněny nebo vyčerpán stanovaný počet kroků, opakuj 2.
- Řeší instance 2 SAT V čase $O(n^2)$ (Papadimitriou 1992)
- Nepracuje dobře na 3 SAT

## Zaujatá náhodná procházka
(Biased Random Walk)

- Algoritmus:
  1. Počáteční ohodnocení *Y*: každou proměnnou ohodnoť 0 nebo 1 se stejnou pravděpodobností.
  2. S pravděpodobností $0<q<1$ proveď 3 jinak proveď 4.
  3. Najdi ohodnocení *Y'*, které se liší od *Y* v právě jedné proměnné některé nesplněné klauzule
  4. Najdi ohodnocení *Y'*, které se liší od *Y* v právě jedné proměnné a poskytne nejvíce splněných klauzulí
  5. $Y \leftarrow Y'$. Pokud nejsou všechny klauzule splněny nebo vyčerpán stanovaný počet kroků, opakuj 2.

## Výhody randomizovaných algoritmů

- strukturní jednoduchost
- očekávaná kvalita výsledku může být lepší než zaručená kvalita aproximativních algoritmů
- nezávislým opakováním se dá zlepšit kvalita

Kombinace s deterministickými prvky: poskytuje nestrannost při vzorkování libovolného souboru prvků:

- každý náhodný start je stejně pravděpodobný
- každý náhodně vybraný sousední stav je stejně pravděpodobný
- každý náhodně vybraný krok z množiny, pro které dává heuristická funkce stejnou hodnotu atd.

## Meze randomizace

- Nechť je čas výpočtu randomizovaným algoritmem $T(n)$.
- Je možno (deterministicky) zkonstruovat posloupnost čísel takovou, že ji žádný algoritmus v čase $T(n)$ nerozezná od náhodné posloupnosti.
- Tudíž, původní randomizovaný algoritmus bude fungovat stejně.

$\Rightarrow$ randomizované paradigma výpočtu není silnější než deterministické

$\Rightarrow$ derandomizace algoritmů