

# Metoda Monte Carlo - Cvičení V (paralelka II) a domácí úlohy

## 1 Další stochastické algoritmy - domácí úkol

### Příklad 1 - Testování prvočíselnosti, 5 bodů, do jednoho týdne (do pondělí)

Prvočíselnost můžeme testovat různými způsoby. Jednou z metod je Millerův-Rabinův test.

1. Implementujte Millerův-Rabinův test pro testování prvočíselnosti. Hodnotu parametru  $m$  (maximální počet generovaných čísel) můžete volit rovnu 100 (popř. jinak - ale pak uveďte svou volbu ve zprávě).
2. Implementujte jiný algoritmus pro testování prvočíselnosti. Například metodu hrubé síly ze skript, kdy se testuje dělitelnost čísla  $n$  všemi čísli mezi 2 a  $\sqrt{n}$ . V případě jiného algoritmu ho ve zprávě stručně popište.
3. Vygenerujte náhodně 100 lichých čísel z intervalu 10000 a 100000 (pokud jste tak nezískali alespoň 10 prvočísel, doplňte sadu o další prvočísla ze stejného intervalu, aby jich bylo alespoň deset). Otestujte oběma metodami, zda se jedná o prvočísla (toto bude program vypisovat - např. do souboru). Výsledky shrňte ve zprávě - kolik bylo mezi čísli prvočísel, úspěšnost algoritmů při jejich detekci. Do zprávy vložte tabulku s časovými nároky obou algoritmů: průměrný čas a jeho směrodatná odchylka pro prvočísla, čísla složená a pro všechna dohromady. V případě, že by dosažené časy byly příliš malé, otestujte na větších číslech. Pro Millerův-Rabinův test v tabulce uveďte i počet iterací (potenciálních „svědků“, než byl nalezen výsledek pro prvočísla, neprvočísla, všechny) - opět průměrná hodnota a odchylka. Shrňte výsledky vlastními slovy (čas v závislosti na algoritmu a na tom, zda se jedná o prvočísla nebo ne).

Pošlete zprávu (ve jednom z formátů .doc, .docx, .odt, .pdf, .xls, .xlsx, .ods či .txt) a zdrojové soubory programu (popř. i jeho výstupy). Povolené programovací jazyky jsou C, C++, Java, C#, Matlab (R2010), Pascal, Python.

### Příklad 2 - Simulované žihání, 10 bodů, do dvou týdnů (do pondělí)

Uvažujeme úlohu TSP (problém obchodního cestujícího), kde graf tvoří 100 uzlů na mřížce  $10 \times 10$  (ekvidistance je 1). Váhy hran mezi jednotlivými uzly jsou dané jejich euklidovskou vzdáleností.

1. Řešte tuto úlohu metodou simulovaného žihání. Můžete volit následující parametry (popř. jiné - podle výkonu Vašeho počítače):  $K = 10000$  je počet opakování při každé teplotě,  $T_{poc} = 5$  je počáteční teplota,  $T_{kon} = 0.01$  je dolní mez pro teplotu,  $\alpha = 0.99$ . Jako počáteční permutaci volte  $\pi(i) = i, i = 1, \dots, 100$ . Volbu všech parametrů uveďte ve zprávě. Program by měl umět vypsát energii průběžných řešení a výsledné řešení. Ve zprávě vhodně vizualizujte nalezené řešení a průběh vývoje energie (pro toto řešení). Shrňte výsledky vlastními slovy.
2. Opakujte výpočet alespoň desetkrát a ve zprávě uveďte střední, minimální a maximální hodnotu energie výsledných řešení a porovnejte je s energií optimálního řešení.
3. Zkuste jiné hodnoty parametru  $\alpha$  (např.  $\alpha = 0.8, \alpha = 0.5$ ) pro vhodné  $K$ . Opakujte výpočet pro každé  $\alpha$  alespoň desetkrát a ve zprávě uveďte střední, minimální a maximální hodnotu energie výsledných řešení a porovnejte je s energií optimálního řešení. Zhodnoťte vlastními slovy zjištěné chování a časové nároky algoritmu v závislosti na volbě parametru  $\alpha$ .

Pošlete zprávu (ve jednom z formátů .doc, .docx, .odt, .pdf, .xls, .xlsx, .ods či .txt) a zdrojové soubory programu (popř. i jeho výstupy). Povolené programovací jazyky jsou C, C++, Java, C#, Matlab (R2010), Pascal, Python.