# Cvičení z Úvodu do aproximačních a pravděpodobnostních algoritmů

ZS2324 - 6. cvičení

## 1 Opakování 1

Férová mince je $n$-krát hozena. Nechť $X_{ij}$ je pro každé $1 \leq i < j \leq n$ indikátorová veličina pro jev „$i$-tý a $j$-tý hod dopadly stejně“. **Ukažte, že $X_{ij}$ jsou po dvou nezávislé, ale nejsou nezávislé.**

## 2 Opakování 2

Nechť $X_1, \ldots, X_k$ jsou diskrétní nezávislé veličiny s rozdělením $U(1, 4)$. (Tedy $P(X_i = j) = 1/4$ pro všechny $1 \leq i \leq k$ a $1 \leq j \leq 4$.)

**Zkonstruujte co nejvíce 2-nezávislých veličin s rozdělením $U(1, 4)$.**

## 3 Derandomizace

Mějme instanci MaxSATu, kde každá klauzule obsahuje nejvýše tři literály. Z přednášky víme, že Best-SAT je alespoň 0.75-aproximační algoritmus. Je ale randomizovaný. **Popište, co potřebujete k jeho derandomizaci.**

Hint: Dokazovat použitelnost podmíněné střední hodnoty je moc práce, zkuste to jinak.

Hint: Co kdyby byly klauzule délky nejvýše 2?

## 4 Počítání různých

Bude nám postupně přečten seznam $n$ čísel. Naším úkolem je spočítat $k$, což je počet unikátních hodnot v tomto seznamu.

Mohli bychom si seznam setřídit. Budeme ale potřebovat schopnost editovat ten seznam, nebo $\Omega(k)$ paměti. Rychlé to taky nebude. ($\Omega(n \log k)$ času.) My se zaměříme na situaci, kdy máme paměti málo. Této oblasti se obecně říká „streaming“ algoritmy, neboli algoritmy pro datové toky.

Pro zjednodušení budeme předpokládat, že máme přístup k dokonale náhodným hashovacím funkcím které si nepotřebujeme pamatovat. Taky se nám bude hodit následující pozorování:

Mějme $k$ uniformně náhodných čísel $X_1, \ldots, X_k$ z intervalu $[0, 1]$. Náhodná veličina $min(X_1, \ldots, X_k)$ má střední hodnotu $1/(k + 1)$ a směrodatnou odchylku $1/(k + 1)$.

1. **(rozcvička) Pokud máme alespoň $\Omega(k)$ paměti, tak by mělo být snadné najít řešení v $\Omega(n)$ času.** **Hint: Použijte randomizaci.**
2. **Naštěstí umíme $k$ alespoň odhadnout i s extrémně málo paměti. Co bychom mohli udělat, když máme jen $\log n$ bitů paměti?**
3. **Co když máme jenom $O(\log \log n)$ bitů?**

## HW9: Parallelní max-4-cut

Vytvořte paralelní approximační algoritmus pro max-4-cut. Máte k dispozici $n^4$ processorů. Cílem je deterministická $\frac{3}{4}$-approximace v čase nejhůře O(log n), ale za horší výsledky bude alespoň část bodů.

HINT: Já bych začal randomizovaným algoritmem, ale nejspíše to jde i jinak.

## HW10: Rovnost multisetů

Multiset je zobecněním množiny, u kterého je povolen vícenásobný výskyt prvků. Tedy např. $\{1, 2, 3\}$ a $\{1, 1, 2, 3\}$ jsou dvě různé multimnožiny.

Jedna z možností, jak rozeznávat rovnost multisetů, je oba multisety setřídit a následně zkontrolovat rovnost prvek po prvku. Tohle ale vyžaduje $\Omega(n \log n)$ času pro multimnožiny s $n$ prvky.

Využijte hashování ke konstrukci Monte Carlo nebo Las Vegas algoritmu pro testování rovnosti multisetů. Cílem je potřebovat jen $O(n)$ prostoru a (pro Las Vegas v průměru) $O(n)$ času. Uveďte, který algoritmus jste sestrojili, a ukažte, že funguje!

## Informace

- Celkem bude zadáno deset úkolů, každý za čtyři body.
- Na vypracování máte dva týdny.
- Potřebujete alespoň 25 bodů na zápočet.
- Domácí úkoly odevzdávejte přes Sovu.

Link: `https://kam.mff.cuni.cz/owl/c/zs2324/apxr/`
Enroll token: 6de8d9714087