2. CVIČENÍ Z ÚVODU DO APROXIMACÍ

pravděpodobnost v rukou informatiky

PŘÍKLAD PRVNÍ **Rocvička s žetony.** Máme deset žetonů s hodnotami 1 až 10. Když je rozdělíme na dvě skupiny zvlášť liché a zvlášť sudé, bude průměr skupin 5 pro liché a 6 pro sudé. Je možné žetony přeuspořádat do dvou jiných skupin tak, že se průměr obou skupin ostře zvýší? Pokud ne, tak to zdůvodněte – a pokud ano, tak rozdělení ukažte.

PŘÍKLAD DRUHÝ **Karetní triky.** Příklad jste už slyšeli na jedné z předchozích přednášek: máme 52 karet (polovina červených, polovina černých) zamíchaných náhodně (rovnoměrně náhodná permutace). Pak odkrýváme karty jednu po druhé. Před každou kartou máte možnost říci „tuhle kartu chci“. Když ji chcete, tak ji odkryjeme a pokud je červená, vyhrajete. Zajímá nás pravděpodobnost, že váš algoritmus vyhraje.

1. Jaká je pravděpodobnost výhry algoritmu $Pr \equiv$ „vždy zkusím první kartu“? A co $Po \equiv$ „vždy zkusím poslední kartu“?
2. Zkusme přijít na „lepší“ algoritmus, než naivní Po. „Lepší“ teď myslíme v následujícím smyslu: nalezněte algoritmus L, který má pravděpodobnost vyšší než $1/2$, že ukáže na červenou, když Po ukáže na černou. Jaký je zásadní problém s touto relací?
3. Bez nutnosti přesného výpočtu zdůvodněte, jakou má nevýhodu následující algoritmus: „Počkám si, než odkryté karty budou mít víc černých karet než červených; pak ve zbytku bude více červených než černých, a pravděpodobnost bude na mé straně (větší než $1/2$). Protože střední hodnota počtu červených karet v sudém počtu karet je 0, pravděpodobnost, že aspoň v jednom kroku se nevyrovnanost otočí na mou stranu, je dost vysoká. A když se pro mne výhodná situace nestane, ukážu na poslední kartu, čili musím být lepší než Po.“
4. Náš hlavní úkol: nalezněte algoritmus, který zvolí červenou kartu s větší pravděpodobností, než Pr nebo Po, nebo dokažte, že žádný takový algoritmus neexistuje.

PŘÍKLAD TŘETÍ **Maximální splnitelnost.** V maximalizačním problému MAX-SAT máme na vstupu formuli v CNF formě $((x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge \ldots)$. Předpokládejme, že každá klauzule má alespoň dva literály. Našim úkolem není splnit celou formuli, ale maximalizovat počet klauzulí, které jsou splněny.

1. Uvažte algoritmus A: Vezměme dvě přiřazení: jedno nastaví všechny proměnné na 1, druhé všechny proměnné na 0. Vrátíme pak to přiřazení, které splnilo všechny klauzule. Jaký je aproximační poměr tohoto (deterministického) algoritmu?
2. Zesilme A takto: uvažme libovolný algoritmus B takový, že vyzkouší konstantně mnoho dopředu připravených přiřazení (nelze je zvolit podle formule) a vrátí nejlepší možné. Přiřazení je libovolná funkce $p : \mathbb{N} \to \{0, 1\}$ taková, že $x_i := p(i)$ (takže jde použít na každou formuli). Jaký je maximální aproximační poměr, kterého algoritmus B může dosáhnout?
3. Co když to zkusíme rovnoměrně náhodně? Spočtěte střední hodnotu počtu splněných klauzulí v tomto případě.
4. Je MAX-k-SAT stejně těžký jako k-SAT? Není. Ověřte nejprve, že 2-SAT (na vstupu je CNF formule s nejvýše dvěma literály na klauzuli, máme rozhodnout, je-li splnitelná) polynomiálně řešitelný problém.
5. Ukažte, že MAX-2-SAT (Na vstupu je číslo c a CNF formule s nejvýše dvěma literály na klauzuli, máme rozhodnout, je-li alespoň c klauzulí splněno) je NP-těžký.