3. CVIČENÍ Z ÚVODU DO APROXIMACÍ

pravděpodobnost v rukou informatiky

PŘÍKLAD PRVNÍ **Max-k-řez.** V problému MAXIMÁLNÍ k-ŘEZ dostaneme na vstupu neorientovaný graf G a váhy na hranách $w\colon E(G) \to \mathbb{R}^+$. Cílem je rozdělit všechny vrcholy grafu do k hromádek $V_1, V_2, \ldots, V_k$ tak, aby součet všech mezihromádkových hran byl co největší možný.

Navrhněte a zanalyzujte pravděpodobnostní $\frac{k-1}{k}$-aproximační algoritmus pro MAXIMÁLNÍ k-ŘEZ.

PŘÍKLAD DRUHÝ **Karetní triky.** 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. Zkuste si po krátké úvaze (bez nutnosti přesného výpočtu) tipnout, 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 $((x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge \ldots)$. Našim úkolem není splnit celou formuli, ale maximalizovat počet klauzulí, které jsou splněny.

1. Co když zkusíme nastavovat proměnné rovnoměrně náhodně? Spočtěte střední hodnotu počtu splněných klauzulí v tomto případě. Jaká by byla střední hodnota/aproximační poměr, pokud bychom předpokládali, že každá klauzule má alespoň dva literály?
2. Je MAX-k-SAT stejně těžký jako k-SAT? Jak už jsme si řekli dříve, 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.
3. 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.
   Hint: Existuje i přímý převod z 3-SAT na MAX-2-SAT. Vezměte klauzuli například s třemi literály a postavte z ní několik (cca 10) nových 2-SAT klauzulí. Cílem je toto: pokud by algoritmus uplněl splnit c z nich, tak už musí splnit původní klauzuli také.
   Druhý hint: Budete potřebovat nějakou další proměnnou, než jen tři z klauzule. Zkuste zadefinovat proměnnou d s významem „klauzule je splněna“. Těžké bude zákodovat tento význam do klauzulí velikosti nejvýš dvě ...