APX. A ONLINE ALGORITMY – DOMÁCÍ ÚKOLY 3

Deadline: čtvrtek **14. 6. 2018 23:59 AoE**. (AoE znamená *Anywhere on Earth*, tedy kdekoliv na Zemi. Najděte si, kdy to přesně je)

Úkoly odevzdávejte nejlépe elektronicky (ve formátu PDF, ODT, ... nebo i jen jako text emailu) Martinu Böhmovi na email `bohm+hw3@iuuk.mff.cuni.cz`. Naskenované či kvalitně vyfocené papíry jsou též přijímány, pokud lze vše přečíst bez problémů. Nebojte se napsat, pokud nějaké zadání není jasné nebo vám přijde, že v něm něco chybí.

První domácí úkol — **MARK není H_k-kompetitivní** [4 body]

Najděte příklad, který ukazuje, že randomizovaný algoritmus MARK pro stránkování (paging) není H_k-kompetitivní. Stačí uvažovat případ $k = 2$ a $n = 4$ (tedy 2 stránky v cachi a 4 stránky celkem), ale je potřeba vytvořit libovolně dlouhou posloupnost, aby důkaz fungoval pro libovolnou aditivní konstantu v definice kompetitivního poměru.

Druhý domácí úkol — **Pravděp. prohledávání přímky** [$R \leq 8$? $\lceil 9 - R \rceil : 0$ **bodů**]

Navrhněte R-kompetitivní pravděpodobnostní algoritmus pro prohledávání přímky pro co nejmenší R. Všimněte si, že váš algoritmus musí být 8-kompetitivní nebo lepší.

Třetí domácí úkol — **Neaproximovatelnost Max 3-Většiny** [4 body]

Dokažte, že neexistuje žádný $\frac{2}{3} + \varepsilon$-aproximační algoritmus pro problém Max 3-Většiny, pokud $P \neq NP$. V tomto problému máme na vstupu seznam podmínek nad Booleovskými proměnnými $x_1, x_2, \ldots, x_n$. Každá podmínka sestává z jedné trojice literálů (třeba $\{x_2, \neg x_5, x_{11}\}$) a význam každé z nich je „tato trojice musí mít většinu literálů rovna 1“. Cílem je maximalizovat počet splněných podmínek.

Tip: Zopakujte si L-redukce a mezeru-uchovávající redukce (např. věta 16.22 ve Williamson, Shmoysovi; strana 422).

Čtvrtý domácí úkol — **Chytání poloroviny ve 2D** [4 body]

Mějme následující online problém v $\mathbb{R}^2$ s klasickou Euklidovskou metrikou (říkejme mu Chytání poloroviny):

Začneme v libovolném bodě $x_0 \in \mathbb{R}^2$. Pak přijde online polorovina P_1 zadaná svou hraniční přímkou p_1. Našim úkolem je okamžitě se posunout do libovolného bodu x_1 ležícím uvnitř P_1 (přímku p_1 chápeme také jako součást poloroviny). Pak dostaneme online polorovinu P_2 a pokračujeme dále; opět se musíme posunout dovnitř P_2, ale už nemusíme být uvnitř P_1 – starých polorovin už si nevšímáme.

Až instance skončí, tak změříme celkovou délku námi procestovaných vzdáleností $\sum_{i=0}^{n-1} d(x_i, x_{i+1})$ a naším úkolem je minimalizovat ušlou vzdálenost (a srovnat se s optimem, které začíná ve stejném bodě a zná všechny poloroviny dopředu, ale stejně je musí navštívit v pořadí).

Pořádně dokažte, že následující dva algoritmy nejsou c-kompetitivní pro žadnou konstantu c:

1. *Hladový algoritmus:* V kroku j se posuneme do nejbližšího bodu uvnitř P_j.
2. *Posun do lokálního optima:* V kroku j si přepočítáme, kam by se posunul optimální algoritmus (který má na vstupu jen $P_1, \ldots, P_j$), a posuneme se do stejného místa.