

Úloha č. 2

Bezpečnostní kamery



Zamysli se!

10 b

Tato úloha je čistě teoretická, tvým úkolem zde není napsat program. Namísto toho si dej záležet na kvalitním slovním popisu, kde mimo jiné jasně zdůvodníš, proč tvůj postup skutečně bude fungovat.

Na svém todo-listu máš ještě jeden důležitý úkol, který je potřeba vyřešit před otevřením ZOO. Před pár týdny zde externí firma Clouda instalovala sofistikovaný kamerový systém KAM, který má zajistit bezpečnost pro všechna zvířata i návštěvníky. V ZOO se celkem nachází $N - 1$ kamer, z nichž každá má přiřazeno jedno unikátní celé číslo mezi 2 a N , které slouží jako její identifikátor. Všechny kamery jsou propojeny kabelovou sítí, skrze kterou by měly být schopné živě odesílat zaznamenávaný obraz na centrální server. Ten je dále propojen s počítačem, který bude obsluhovat ochranka a vše sledovat. Bohužel se zdá, že instalace kabelů se firmě úplně nepodařila – tyto kabely totiž podporují jen jednosměrnou komunikaci, a tak jsme schopní posílat data pouze z centrálního počítače do jednotlivých kamer, ale ne opačně. Komunikace je navíc extrémně pomalá, každý kabel zvládne poslat pouze jeden bit za sekundu. Protože je pátek večer, do firmy se ti nedaří dovolat. Začneš tedy KAM prozkoumávat na vlastní pěst.

Shodou náhod jsou kabely nainstalovány poměrně strukturovaným způsobem: Pokud centrální server označíme číslem 1 a každou kameru jejím unikátním identifikátorem, pak pro každou dvojici uzlů $i, j \in \{1, \dots, N\}$ v síti platí, že kabel z i do j vede (a tedy lze z i do j posílat data) právě tehdy když číslo i beze zbytku dělí číslo j .

Kromě jiného se ti podařilo najít i krátký návod ke KAMu. V návodu se dočítáš, že všechna zařízení v systému KAM podporují i bezdrátovou komunikaci! Každé zařízení umí vysílat pomocí radiových vln. Pokud jsou však dvě zařízení blízko sebe, je potřeba, aby každé vysílalo na jiné frekvenci, jinak se vlny vzájemně vyruší. Zařízení, která jsou blízko sebe, jsou typicky také propojena kabelem, takže by bylo vhodné, aby každá dvě propojená zařízení vysílala na odlišné frekvenci. V návodu se dále píše, že pokud při počáteční konfiguraci byl do paměti každého zařízení uložen jeho správný identifikátor i , pak si každé zařízení po svém zapnutí automaticky a okamžitě nastaví vysílací frekvenci tak, aby k žádnému rušení nedocházelo. Při přečtení této informace se na několik minut pozastavíš. Skutečně je něco takového možné?

Část 1 (4 body)

Pro zjednodušení předpokládejme, že frekvence je libovolné přirozené číslo (ve skutečnosti jsou frekvence reálná čísla, ovšem pokud je chceme volit tak, aby se vzájemně nerušily, nabízí se volit z posloupnosti předem určených čísel, mezi nimiž necháme dostatečně dlouhé prázdné intervaly – v našem případně budou mít příhodně délku 1). Popiš algoritmus, pomocí kterého může každé zařízení určit svou frekvenci tak, aby žádná dvě zařízení, mezi kterými vede kabel, nezvolila stejnou frekvenci, a zároveň aby počet různých frekvencí byl nejmenší možný. Dokaž, že Tvůj algoritmus je skutečně optimální, tedy že pro každé N je nemožné docílit menšího počtu různých frekvencí. Algoritmus bude spuštěn paralelně na všech zařízeních současně. Na vstupu dostane pouze jedno číslo $1 \leq i \leq N$, tedy identifikátor daného zařízení. Žádné

informace o sousedních zařízeních ani jejich počtu na vstupu *nejsou*. Zařízení musí svou frekvenci nastavit okamžitě po zapnutí, nicméně Tobě prozatím stačí jakýkoliv algoritmus, jehož výpočet zaručeně skončí v konečném čase (každý pomalý algoritmus je přeci možné nějak zoptimalizovat). Zařízení spolu nicméně nemohou žádným způsobem komunikovat, jelikož komunikace přes kabely je inherentně pomalá a zoptimalizovat nepůjde.

Nojo, ale můžeme se skutečně spolehnout na to, že jsou všechna zařízení správně nakonfigurována? V návodu se dále píše, že pokud tomu tak není, je možné z centrálního serveru spustit nouzový mód konfigurace. V tomto módu si opět každé zařízení nastaví vysílací frekvenci tak, aby nebylo v konfliktu s žádným sousedním zařízením. Tentokrát ale ke správné konfiguraci bude potřeba komunikace mezi zařízeními skrze pomalé kabely. V návodu se píše, že celá konfigurace nebude nikdy trvat déle než $\log_2(N) + 5$ sekund.

Část 2 (6 bodů)

Popiš algoritmus, pomocí kterého může každé zařízení určit svou frekvenci tak, aby byly dodrženy stejné podmínky jako v Části 1. Algoritmus tentokrát nedostane na vstupu číslo i . Namísto toho dostane počet vstupních a výstupních kabelů zapojených do zařízení. Navíc je nyní možné posílat přes kabely zprávy do sousedních zařízení. Bohužel však není dostupná informace o tom, do kterých zařízení jednotlivé výstupní kabely vedou. Každé zařízení má vnitřní hodiny, které jsou synchronizované napříč všemi zařízeními. Algoritmus je spuštěn přesně jednou za sekundu, opět paralelně na každém zařízení. Na vstupu nyní dostane dvě čísla p_i (počet vstupních kabelů), q_i (počet výstupních kabelů), a dále všechny příchozí zprávy, které mu byly přes vstupní kabely poslány v minulé sekundě. Na výstupu je seznam zpráv zasláných přes výstupní kabely (maximálně 1 bit na každý kabel; není nutné v každém kole něco odeslat). Hodnoty všech proměnných, které algoritmus nastaví, zůstanou uloženy v paměti a je možné k nim přistoupit i při následujících spuštěních v rámci jednoho zařízení. Opět nám nezáleží na časové složitosti výpočtů (předpokládáme, že náš výpočet určitě stihne doběhnout během jedné sekundy). Nicméně potřebujeme zajistit, že nejpozději po zmíněných $\log_2(N) + 5$ kolech komunikace (tedy sekundách) bude každé zařízení mít uloženu svou vysílací frekvenci ve vnitřní proměnné.