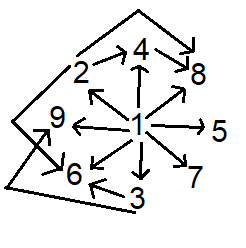
Fiks 2021/2022 úloha č. 14 „Kamery“

**Kryštof Olík**

V ZOO máme celkem *N*-1 kamer, ze kterých každá je označená indexem *i*, které je v rozmezí přirozených čísel { 2, …, *N* }. Index číslo jedna patří centrálnímu serveru. Dále platí, že jsou kamery propojeny jednosměrným kabelem, který vede z kamery s indexem *i* do kamery s indexem *j* pokud *i* | *j* , čili pokud index *i* dělí index *j* beze zbytku. Každá kamera vysílá rádiové vlny ve frekvenci přirozených čísel.



**Grafické znázornění kabelového připojení**

## Část 1

Naším cílem je vymyslet algoritmus, ve kterém kamera vezme svůj index *i* a přiřadí si frekvenci vysílání takovou, aby žádné kamery připojené kabelem k ní neměli stejnou frekvenci. Zároveň je naším cílem, aby počet různých frekvencí byl co nejmenší.

Nejdůležitější je si najít všechny dělitele indexu *i* – ty si označíme písmenem *D*. Například pro *i* = 8 platí, že *D* = { 1, 2, 4, 8 }. Z toho můžeme zjistit všechny zapojené kamery, které vedou do kamery s indexem *i*: to je množina D \ { 1, *i* }. Teď už zbývá jenom zjistit jaké frekvence mají zapojené kamery, ale jak?

Vytvoříme si rekurzivní funkci. První si nastavíme základní frekvenci = 1, ke které budeme postupně přidávat. Pokud D \ { 1, *i* } = { }, tak také frekvence 1 zůstane. Začneme iterovat přes seřazenou množinu D \ { 1, *i* }. Rekurzivně zavoláme tuto samou funkci, abychom zjistili frekvenci iterované kamery. Dále postupuje takto: pokud se poslední frekvence předešlé kamery nerovná té zjištěné, tak přičteme ke frekvenci 1. V pseudokódu by se tato funkce napsala takto:

def frekvence\_kamery(i):

predesle\_kamery = D \ { 1, *i* }

frekvence = 1

posledni\_frekvence\_predesle\_kamery = 0

for kamera in predesle\_kamery:

frekvence\_predesle\_kamery = frekvence\_kamery(kamera)

if (posledni\_frekvence\_predesle\_kamery != frekvence\_predesle\_kamery):

frekvence += 1

posledni\_frekvence\_predesle\_kamery = frekvence\_predesle\_kamery

return frekvence

Na konec nám zbývá dokázat, že algoritmus opravdu najde pro všechny kamery frekvenci tak, aby bylo různých frekvencí co nejméně, a že je algoritmus konečný. Díky tomu, že rekurzivně postupujeme a zjišťujeme všechny frekvence napojených kamer, si můžeme být jistí, že opravdu najdeme nejmenší možný počet frekvencí. Algoritmus je konečný - sice se ve funkci nachází rekurze, ale ta se pokaždé zastaví poté co dojde na index, u kterého platí D \ { 1, *i* } = { }.

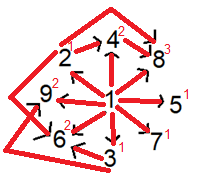
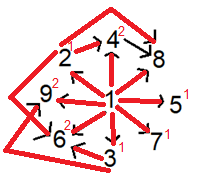
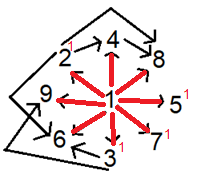
Tímto jsme vytvořili algoritmus, díky kterému si může každá kamera, která má v paměti uložený svůj index, nastavit správnou frekvenci.

## Část 2

Ve druhé části musíme splnit stejné podmínky jako v první, čili každé kameře přiřadit frekvenci takovou, aby nerušila kamery připojené na ní, a zároveň, aby různých frekvencí bylo co nejméně. Je tu však jeden háček. Tentokrát kamery nemají uložený svůj index, ale mají uložený počet vstupních a výstupních kabelů. Zároveň můžeme každou sekundu posílat přes kabelovou sít bit.

První sekundu vyšle centrální počítač bit do všech kamer. Kamerám, které mají pouze jeden vstupní kabel, nastavíme frekvenci 1 (o kolikáté kolo/sekundu se jedná). Dále z těchto kamer vyšleme bit do dalších kamer. Kamery, do kterých přišel počet bitů, jako jejich počet vstupů, vyšlou bit do všech výstupů a sobě nastaví frekvenci stejnou jako momentální kolo. Toto opakujeme do té doby, dokud nemá každá kamera svou frekvenci.

Pro lepší představu je tady graficky znázorněn příklad, kde N = 9:



Červená znamená kabel, do kterého už byl poslaný bit. A červené číslo je nastavená frekvence kamery.

Ještě nám zbývá dokázat, že algoritmus opravdu funguje. Jelikož je kamera závislá na všech vstupních kamerách, které společně s informací času dávají kameře všechny potřebné informace, tak se nemůže stát, že by se nastavila špatná frekvence – nikdy se nenastaví stejná frekvence jako frekvence sousedů. Můžeme se kouknout na jakýkoliv příklad a uvidíme, že poslední signál dojde do kamery opravdu v ten správný čas. A jelikož signál se zasílá dál ihned po všech vstupních signálech, tak si můžeme být jistí, že bude frekvence nejmenší možná.