Bezpečnostní kamery

Daniel Skýpala

13. dubna 2022

Zamyšlení

Začněme tím, že celá situace odpovídá grafu: Kamery jsou vrcholy a kabely jeho hrany. Pro tento grafu hledáme obarvení (přiřazení frekvencí = barev), tak aby dva vrcholy spojené hranou neměli stejnou barvu.

Nejdřív si vezmeme nějaký vrchol i, a podívejme se, s jakými dalšími vrcholy je spojen. Na to si rozepišme vrchol i jako jeho prvočíselný rozklad:

$$i = 1 \cdot p_1 \cdot p_2 \cdots p_n$$

Kde platí, že p_i je prvočíslo a také $1 \leq p_1 \leq p_2 \leq \cdots \leq p_n$. Prvočíselný rozklad budu nyní značit (pozor, tohle je multimnožina, a pokaždé když dále zmíním podmnožinu, tak ve skutečnosti záleží i na počtu prvků):

$$P(i) = \{1, p_1, p_2, \ldots, p_n\}$$

Nyní si rozmysleme, co platí, když vede hrana z i do j.

$$ki = j$$

Rozložíme:

$$(1 \cdot p_{k1} \cdots p_{kn})(1 \cdot p_{i1} \cdots p_{in}) = (1 \cdot p_{j1} \cdots p_{jn})$$

A to sloučíme (pozor, jednička je vlevo dvakrát):

$$P(k) \cup P(i) \setminus \{1\} = P(j)$$

$$P(i) \subseteq P(j)$$

Budeme potřebovat si ale dokázat ještě jednu věc. Konkrétně to, že když máme dva stejně dlouhé rozklady a jeden je podmnožinou druhého, tak jde o rozklady stejného čísla (a opačně):

$$P(i) \subseteq P(j) \ \land \ |P(i)| = |P(j)| \ \Leftrightarrow \ i = j$$

Tak začneme s tím, že když i je rovno j, tak jejich rozklady jsou jistě stejné a mají i stejnou velikost. Zbývá nám ještě implikace zprava doleva. Představme si, že máme dva stejně velké rozklady $P(i), P(j)$ a zároveň víme, že $P(i)$ je podmnožinou $P(j)$. $P(j)$ musí tedy obsahovat všechny prvky z $P(i)$, ale na žádný další nezbývá místo (jinak by nebyly stejně dlouhé). Takže $P(i) = P(j)$, ale protože součin rozkladu je roven číslu, ze kterého jsme ho vytvořili ($i = \prod_{p \in P(i)} p$), tak musí platit i $i = j$.

Nyní máme všechno, co potřebujeme:

Část 1

Jak přiřazovat vrcholům barvy? Určitě vrchol 1 bude mít unikátní barvu (např. 1), protože je spojena se všemi zařízeními. Dvojka bude mít tedy barvu 2. Čtyřku dělí jedna i dva, tedy nejnižší volná barva je 3 a tak dále.

Můžeme si všimnout, že s každou mocninou dvojky se délka rozkladu o jedna prodlouží, takže se nabízí přidělit barvu podle délky rozkladu:

$$f(i) = |P(i)|$$

Proč nyní nemáme dva různé vrcholy i, j spojené hranou se stejnou barvou? Uvažujme, že jsou spojené hranou $P(i) \subseteq P(j)$ a mají stejnou barvu $|P(i)| = |P(j)|$. Ale jak jsme výše ukázali, to nastane právě tehdy, když $i = j$, tedy zařízení nejsou různá. Tedy žádné dva vrcholy spojené hranou nemají stejnou barvu.

A proč je počet barev optimální? Nejdříve se zeptejme, jaká je nejvyšší zvolená barva. To je velikost rozkladu čísla $i < N$, kde i má nejdelší rozklad. Čísel s nejdelším rozkladem, může být více, ale rozmysleme si, že když máme nějaké, můžeme provést následující věc: Všechna prvočísla (ne jedničku) v rozkladu nahradíme dvojkou. Tím jsme jednak nezměnili barvu (protože délka rozkladu je stejná) a zároveň jsme se nedostali nad N, protože prvočísla jsme nahradili za dvojky (které nejsou větší). (A když jsou menší činitelé, je i menší součin.) Tedy největší barvu v grafu bude mít mocnina dvojky. A o té víme že:

$$P(2^k) = \{1, 2_1, 2_2, \ldots, 2_k\}$$

$$|P(2^k)| = k + 1$$

Pro velikost grafu N bude maximální počet barev tedy $\lfloor \log_2 N \rfloor + 1$. A proč je tento počet barev optimální? Všimněme si, že mocniny dvojky (včetně jedničky) tvoří úplný podgraf (každou mocninu dělí všechny nižší: $\frac{2^a}{2^b} = 2^{a-b}$, což je zřejmě celé číslo pro $a > b$), který má velikost $\lfloor \log_2 N \rfloor + 1$ (mocnin dvojky je $\log_2 N$ a k tomu ještě jednička). A protože každá barva v úplném podgrafu může být jen jednou (jinak by vrcholy se stejnou barvou byly spojené hranou), tak určitě nemůžeme mít méně jak $\lfloor \log_2 N \rfloor + 1$, tedy naše obarvení je optimální.

Algoritmus, který použijeme pouze zjistí, kolik má daný identifikátor prvočíselných dělitelů a podle toho vrátí barvu. To se dělá zkoušením čísel od 2 do i, a když dané číslo dělí identifikátor, tak ho vydělíme a připočteme jedna k barvě. (Více to rozebírat nebudu, protože se jedná o velmi známý algoritmus, a kdybychom chtěli být matematici, tak se ho můžeme snažit i zrychlit.)

```
kamera.frekvence = 1
id = kamera.identifikátor
dělitel = 2
dokud (id != 1):
    když id mod dělitel == 0:
        id /= dělitel
        kamera.frekvence++
    jinak:
        dělitel++
```

Časová složitost je v tomto případě $O(id) = O(N)$. (Můžeme si snadno rozmyslet, že v každe iteraci cyklu buď klesne se zvýší dělitel o 1 - a ten může být větší než id jen v poslední iteraci cyklu, nebo se id zmenší alespoň na půl.) Paměťová složitost je jen z pohledu na program $O(1)$.

(Pozor - algoritmus je lineární k číslu N, ale exponenciální vzhledem k jeho binárního zápisu. - Takže se ve skutečnosti jedná o exponenciální algoritmus, který je děsně pomalý, jakmile nám naroste N.)

Část druhá

Pro řešení druhé části využijeme následujícího tvrení:

$$\forall i \in \mathbb{N} \wedge i > 1 : \exists j \in \mathbb{N} \wedge i \neq j \wedge P(j) \subset P(i) \wedge |P(j)| + 1 = |P(i)|$$

Neboli pro každý vrchol $i > 1$ existuje jiný vrchol j takový, že mezi i a j vede hrana a j má o jedna kratší rozklad (tedy i o jedna menší frekvenci) než i. Proč tohle platí? Představme si rozklad čísla i a vyškrtněme z něj prvočíslo. Tím jsme dostali o jedna kratší rozklad, který je zároveň podmnožinou původního. A tento rozklad reprezentuje číslo j.

Na obarvování půjdeme tak, že budeme obarvovat stejně jako v první části, akorát to musíme udělat z jiných informací. Všimneme si, že všechny vrcholy (budu jim říkat předchůdci), ze kterých vede kabel do aktuálního vrcholu, mají menší barvu (Podmnožina je menší než multimnožina). Pokud v i-té ($i \in \mathbb{N}_0$) sekundě budeme obarvovat barvou $i+1$, tak určitě když dojde řada na aktuální vrchol, tak všichni jeho předchůdci budou v tu chvíli obarvení. Zároveň všichni jeho předchůdci budou obarvení právě o sekundu dříve, protože jeden z nich má o jedna menší rozklad, tedy i barvu a bude tedy obarven v $i-1$. sekundě.

Z toho se nabízí následující algoritmus - v každé sekundě si v kabelech budeme posílat bit, jestli je vrchol, ze kterého vede kabel už obarvený. V tu chvíli, kdy zjistíme, že všichni předchůdci už obarvení jsou, nastavíme současnému vrcholu barvu čas + 1, což nastane o 1 sekundu potom, než obarvíme posledního z předchůdců (kvůli zpoždění v přenosu kabelů).

Důkaz o tom, že barva i-tého vrcholu je stejná jako v prvním případě, tedy $f(i) = |P(i)|$, provedeme indukcí:

1. Vrcholu 1 to přiřadí barvu $|P(1)| = 1$. V nulté sekundě spustíme program a vrchol 1 je jediný, který nemá vstupní kabely. (Ostatní vrcholy jsou děleny

beze zbytku jedničkou.) To znamená, že ve všech příchozích kabelech bude jednička (a co, že žádný takový není?). Tedy v čase 0 přiřadíme vrcholu 1 barvu $t + 1 = 0 + 1 = 1$.

2. Pokud přiřadíme správnou barvu všem vrcholům s menším číslem než současnému vrcholu i, tak přiřadíme správně barvu i vrcholu i. Jak jsem zmínil, všichni předchůdci vrcholu i jsou menší a mají i menší barvu než i (protože podmnožina je menší a musí mít i menší součin o ty prvky, které neobsahuje).

   Pokud má vrchol i mít barvu $|P(i)|$, tak (jak jsme dokázali) existuje jeho správně obarvený předchůdce j, který má barvu $|P(i)| - 1$. Tedy nejnižší čas, kdy lze obarvit vrchol i, je kvůli zpoždění kabelů o jedna větší než čas, kdy jsme obarvovali vrchol j. Zároveň, protože všichni předchůdci mají menší barvu, tak v tu chvíli, kdy je obarvený vrchol j jsou taky obarvení (protože jinak by neměli barvu podle předpokladu). Náš vrchol tedy bude obarven v čase o 1 později než vrchol j:

   $$f(i) = t_i + 1 = t_j + 2$$

   A protože barva vrcholu j je určena podle času:

   $$f(j) = t_j + 1$$

   A vrchol j je obarvený správně a má o jedna menší rozklad než i:

   $$|P(i)| - 1 = t_j + 1$$

   Z toho:

   $$f(i) = |P(i)|$$

   Takže jsme obarvili vrchol správně.

Vzhledem k tomu, že obarvujeme stejně, jako v části jedna, tak správnost a optimálnost obarvení jsme již dokázali. Zbývá rozmyslet si čas - Použijeme to, že nejvyšší barva v grafu je $\lfloor \log_2 N \rfloor + 1$, a ta byla obarvena v přesně daný čas.

$$f(i) = t + 1$$

$$\lfloor \log_2 N \rfloor + 1 = t + 1$$

$$t = \lfloor \log_2 N \rfloor$$

To znamená, že po $\lfloor \log_2 N \rfloor + 1$ sekundách (pozor - nultá sekunda se taky počítá), budou všechny barvy nastavené. Pokud bychom to chtěli uvnitř kamer rozpoznat, tak můžeme zkontrolovat jen čas a v $\lfloor \log_2 N \rfloor + 1$. sekundě skončit.

Jeden tik bude vypadat následovně:

```
když kamera.čas == 0: // čtení ze synchronizových hodin
    když kamera.p == 0: // počet vstupních zařízení
        kamera.připravena = 1
```

```
        kamera.frekvence = 1
    jinak:
        kamera.připravena = 0
jinak když kamera.připravena == 0:
    součet = 0
    pro každou zprávu v kamera.zprávy: // seznam všech přijatých zpráv
        součet += zpráva
    když součet == kamera.p:
        kamera.připravena = 1
        kamera.frekvence = kamera.čas + 1

pro každý kabel v kamera.výstupní_kabely: // seznam všech výstupních kabelů
    kabel.pošli(kamera.připravena)
```

Časová složitost jednoho tiku je podle délky cyklů $O(p+q) = O(N)$. Tiků je celkem logaritmicky mnoho, celkem tedy pro přenosovou rychlost P: $O(N \log N + P \log N)$. Paměťová složitost, protože příchozí a odchozí informace nezapočítáváme, je $O(1)$.

Ještě se nabízí zmínit, že bychom to stihli zvládnout v $\lfloor \log_2 N \rfloor$ sekundách s menším trikem. - V nulté sekundě jsme schopni rozeznat vrchol 1 s barvou 1 (má 0 vstupních kabelů) a prvočísla s barvou 2 (1 vstupní kabel z jedničky) a oba typy obarvovat v 0. sekundě. Každý další vrchol by barvu akorát nastavoval podle $t + 2$, protože jsme první dvě sekundy smrskli do jedné.