73. ročník Matematické olympiády – 2023/2024

Úlohy domácího kola kategorie P

Řešení úloh odevzdávejte pomocí webového rozhraní, které je dostupné na stránkách olympiády *https://mo.mff.cuni.cz/*. Tam také najdete podrobnější instrukce k odevzdávání a další informace o kategorii P.

Úlohy P-I-1 a P-I-2 jsou praktické. Vaším úkolem v nich je vytvořit a odladit efektivní program v jazyce Pascal, C nebo C++. Řešení těchto dvou úloh odevzdávejte přes webové rozhraní ve formě zdrojového kódu. Odevzdaná řešení budou automaticky vyhodnocena pomocí připravených vstupních dat a výsledky vyhodnocení se dozvíte krátce po odevzdání. Pokud váš program nezíská plný počet 10 bodů, můžete své řešení opravit a znovu odevzdat.

Úlohy P-I-3 a P-I-4 jsou teoretické, za každou z nich lze získat až 10 bodů. Řešení musí obsahovat popis algoritmu, zdůvodnění jeho správnosti a v úloze P-I-3 též odhad časové a paměťové složitosti. Nemusíte psát program, algoritmus stačí zapsat ve vhodném pseudokódu nebo dokonce jenom slovně, ale v tom případě dostatečně podrobně a srozumitelně. Hodnotí se nejen správnost, ale také efektivita zvoleného postupu řešení. Řešení obou teoretických úloh odevzdávejte ve formě souboru typu PDF přes výše uvedené webové rozhraní.

Řešení všech úloh můžete odevzdávat do 15. listopadu 2023. Opravená řešení a seznam postupujících do krajského kola najdete na webových stránkách olympiády.

Experimentální jazyky: V letošním ročníku MO-P je také možné odevzdávat praktické úlohy i v jazycích Python 3, C# 11 a Java 11. U nich však nezaručujeme, že bude možné získat plný počet bodů – je možné, že program nestihne doběhnout do časového limitu, byť používá algoritmus s optimální časovou složitostí.

P-I-1 Potrubní pošta

V budově úřadu pro zbytečné otázky je celkem n kanceláří, očíslovaných od 0 do $n-1$. V každé kanceláři sedí úředník, který se specializuje na zbytečné otázky jednoho konkrétního typu. Má k dispozici terminál potrubní pošty, což je zařízení, na kterém je k koncovek pro potrubí. Když koncovky na dvou různých terminálech propojíme potrubím, mohou si pak příslušní úředníci navzájem posílat kapsle se zprávami a tak spolu například konzultovat, když jim někdo položí zbytečnou otázku mimo jejich specializaci. Zatím jsou propojené pouze některé koncovky; v budově je v současnosti dohromady p potrubí.

Dvě kanceláře spolu *mohou komunikovat*, jestliže existuje způsob, jak dostat potrubími zprávu z jedné z nich do druhé, a to případně i nepřímo: Zpráva může postupně projít přes víc potrubí a kanceláří, v nichž ji příslušný úředník přepošle dál jiným potrubím. **Všechna v současnosti existující potrubí jsou nezbytná; t.j. kdybychom libovolné z nich odstranili, kanceláře, které propojuje, by spolu už nedokázaly komunikovat, a to ani nepřímo.**

Nový ředitel rozhodl, že je nutné, aby spolu mohli komunikovat všichni úředníci. Na vás je vymyslet, která potrubí je k tomu potřeba přidat.

Soutěžní úloha

Na vstupu dostanete popis aktuální sítě potrubí. Váš program musí určit, která potrubí přidat tak, aby ve výsledné síti spolu mohly komunikovat (alespoň nepřímo) každé dvě kanceláře. Je možné přidat libovolná potrubí, jediným omezením je, že v každé místnosti je jen k koncovek. Přitom přidaná potrubí nemusí být ve výsledné síti nezbytná. Je možné přidat i úplně zbytečné potrubí s oběma konci ve stejné kanceláři; takové potrubí spotřebuje dvě koncovky.

Formát vstupu

Na prvním řádku jsou tři mezerami oddělená celá čísla n, k a p ($1 \leq n \leq 100\,000$, $0 \leq k \leq 100\,000$ a $0 \leq p \leq n-1$). Každý z p následujících řádků obsahuje dvě celá čísla x a y ($0 \leq x, y \leq n-1$) popisující, že již existuje potrubí mezi kancelářemi číslo x a y. Všechna potrubí na vstupu jsou nezbytná (ve výše popsaném smyslu) a v každé místnosti je použito nejvýše k koncovek.

Formát výstupu

Jestliže žádné řešení neexistuje, vypište jeden řádek obsahující číslo -1.

Jestliže řešení existuje, vypište libovolné z nich v následujícím formátu: Na první řádek vypište počet q nově přidaných potrubí. Na každý z následujících q řádků vypište čísla dvou kanceláří, které chcete propojit, oddělená mezerou. Potrubí je možné vypsat v libovolném pořadí.

Počet potrubí q ve vašem řešení nemusí být nejmenší možný. Přijato bude libovolné správné řešení s $q \leq 500\,000$ (pro každý zadaný vstup, pro který řešení existuje, existuje i nějaké s q výrazně menším než $500\,000$).

Bodování

Je pět sad testovacích vstupů, v následující tabulce se pro každou sadu dozvíte počet bodů a dodatečná omezení na n a k pro všechny vstupy v dané sadě:

č. sady	1	2	3	4	5
body	1	3	2	2	2
omezení	$n = 5$	$2 \leq n \leq 10$	$2 \leq n \leq 100$	$2 \leq n \leq 3\,000$	$1 \leq n \leq 100\,000$
	$k = 4$	$2 \leq k \leq 10$	$1 \leq k \leq 100$	$2 \leq k \leq 3\,000$	$0 \leq k \leq 100\,000$

Příklady

Vstup:	*Výstup:*

`5 2 2`
`0 3`
`1 4`
	`3`
	`0 2`
	`2 1`
	`4 3`

Vstup popisuje pět kanceláří, v každé z nich jsou dvě koncovky. Existují již dvě potrubí, mezi kancelářemi 0 a 3 a kancelářemi 1 a 4. Výstup popisuje řešení, které všechny kanceláře propojí do kruhu, čímž jistě dosáhneme toho, že každé dvě spolu mohou komunikovat.

Existuje i mnoho dalších řešení; například, z výše popsaného řešení je možně jedno libovolné potrubí vynechat.

Vstup:	*Výstup:*

`6 4 4`
`0 1`
`1 2`
`3 4`
`4 5`
	`2`
	`1 4`
	`5 5`

V tomto příkladu by stačilo přidat jedno potrubí spojující jednu z kanceláří 0, 1 či 2 s kanceláří 3, 4 nebo 5. My jsme přidali potrubí mezi kancelářemi 1 a 4 a navíc ještě zbytečné potrubí s oběma konci v kanceláři 5. Toto potrubí ničemu nepomůže, ale jelikož máme dost volných koncovek, ani ničemu neublíží.

Vstup:	*Výstup:*
`9 1 1` `4 7`	`-1`

Jelikož je v každé kanceláři jen jedna koncovka, kanceláře 4 a 7 již nejde s žádnou jinou kanceláří propojit.

P-I-2 Barevný plot

Nikomu to neříkejte, ale naše městečko prý navštíví pan prezident! Samozřejmě mu ukážeme místní světově proslulý barevný plot. Bylo by ho ale jistě lepší předem přemalovat, aby vypadal reprezentativněji. Jelikož vlastníte jediný štětec ve městě, gratulujeme vám ke zvolení do jednočlenného výboru pro revitalizaci plotu.

Soutěžní úloha

K dispozici máte 10^9 různých barev, očíslovaných od 0 do $10^9 - 1$. Plot je tvořen n laťkami, očíslovanými zleva doprava od 0 do $n - 1$. Laťka číslo i má aktuálně barvu f_i.

Do návštěvy prezidenta zbývá d dní. Každý den zvládnete přemalovat jednu laťku. A aby bylo vidět, jak pilně váš výbor pracuje, každý den je **nutné** právě jednu laťku přemalovat na **jinou** barvu než tu, kterou právě má. Není ale nutné, aby se každý den přemalovávala jiná laťka.

Vaším cílem je, aby při návštěvě prezidenta měl celý plot stejnou barvu (je jedno jakou), či pokud to nejde, tak aby alespoň na plotě byl co nejdelší souvislý úsek obarvený celý jen jednou barvou.

Formát vstupu

Na prvním řádku vstupu je číslo $t \in \{1, 2\}$ udávající typ testu. Na druhém řádku jsou celá čísla n a d ($1 \le n \le 250\,000$ a $0 \le d \le 250\,000$) oddělená mezerou. Na třetím řádku je n mezerami oddělených celých čísel $f_0, \ldots, f_{n-1}$ ($0 \le f_i \le 10^9 - 1$ pro $i \in \{0, \ldots, n - 1\}$) udávajících počáteční barvy jednotlivých latěk.

Formát výstupu

Na první řádek výstupu vypište jedno celé číslo udávající délku nejdelšího jednobarevného úseku, který je možné vyrobit. Pro testy s $t = 1$ je důležitý jen tento řádek a na výstupu za ním může následovat cokoliv.

Pro testy s $t = 2$ vypište dále d řádků popisujících libovolný postup malování, kterým tohoto výsledku dosáhnete. Na j-tém z nich vypište dvě celá čísla: Číslo b_j laťky přebarvené j-tý den a číslo g_j nové barvy této laťky. Připomínáme, že nová barva se vždy musí lišit od aktuální barvy laťky.

Bodování

Je deset sad testovacích vstupů, za správné vyřešení každé z nich získáte jeden bod. Sady s čísly 1 až 5 obsahují pouze vstupy s $t = 1$. Sady s čísly 6 až 10 obsahují přesně stejné vstupy, ale s $t = 2$. V následující tabulce uvádíme další omezení platné pro všechny vstupy v jednotlivých sadách:

číslo sady	*omezení*
1, 6	malá čísla barev, $d = 0$
2, 7	malá čísla barev, $2 \le n \le 100$, $1 \le d \le 100$
3, 8	malá čísla barev, $2 \le n \le 5\,000$, $1 \le d \le 5\,000$
4, 9	malá čísla barev, málo času, $2 \le n$
5, 10	žádná omezení navíc

Omezení „malá čísla barev“ znamená, že všechny barvy použité ve vstupu mají číslo menší než 250 000; vaše řešení ale může používat i větší barvy.

Omezení „málo času“ znamená, že ve vstupech z této sady není v zadaném počtu dní možné celý plot přebarvit na jednobarevný.

Příklady

Vstup:	*Výstup:*
2	5
5 2	4 7
7 3 7 7 9	1 7

Jediné optimální řešení je přebarvit celý plot na barvu 7. Toho lze dosáhnout přebarvením latěk 1 a 4 v libovolném pořadí.

Vstup:	*Výstup:*
2	2
5 1	1 30
10 20 30 40 50	

Lze dosáhnout pouze toho, že jsou vedle sebe dvě laťky stejné barvy. Je mnoho různých řešení, ukázkový výstup zajistí, že laťky 1 a 2 mají obě barvu 30.

Vstup:	*Výstup:*
1	5
5 3	trikrat premaluji
7 7 7 7 7	stejnou latku

Jelikož jde o vstup s $t = 1$, důležitý je pouze první řádek výstupu, na zbytku nezáleží.

P-I-3 Policajti a zloděj

Gotham je americké město s pravidelnou sítí ulic: r vedoucích ze západu na východ (těm budeme říkat horizontální) a s vedoucích od severu na jih (těm budeme říkat vertikální). Horizontální ulice jsou očíslované od 0 do $r-1$ od nejsevernější k nejjižnější a vertikální pak od 0 do $s-1$ od nejzápadnější k nejvýchodnější.

V Gothamu honí dva policisté zloděje. Na začátku je každý z nich na jedné z křižovatek. Na dané křižovatce jich může být i víc, a to i později během honičky. Policisté a zloděj se pohybují na střídačku, nejprve policisté, pak zloděj, pak opět policisté, atd. Když se pohybují policisté, každý z nich zvlášť se rozhodne, zda zůstane na místě, nebo se přesune na sousední (horizontálně či vertikálně) křižovatku. Obdobně, když se pohybuje zloděj, může buď zůstat na místě, nebo se přesunout na sousední křižovatku.

Policisté zloděje chytí, když alespoň jeden z nich dojde na křižovatku, na které právě stojí zloděj. Pomozte chytit zloděje!

Soutěžní úloha

Napište program, který bude policistům radit, jak se hýbat. Tento program načte ze vstupu čísla r a s a počáteční pozice policistů a zloděje. Poté střídavě vypisuje na vstup popis pohybu obou policistů a načítá ze vstupu popis pohybu zloděje. Detaily formátu vstupu a výstupu si můžete zvolit, jakkoliv vám budou vyhovovat.

Bodování

Vaše řešení musí implementovat strategii, která zajistí, že policisté zloděje po konečném počtu kroků chytí, jestliže pro danou počáteční pozici taková strategie existuje. Nutnou součástí řešení je dostatečně detailní popis této strategie a **zdůvodnění její správnosti**.

Plných 10 bodů získají řešení efektivní pro $r, s \leq 100\,000$. Nejvýše 6 bodů mohou získat řešení efektivní pro $r, s \leq 8$. Část bodů mohou získat i řešení, která pouze načtou počáteční pozici a rozhodnou, zda policisté mají strategii, která zajistí chycení zloděje, nebo zda zloděj může utíkat navždy.

Příklad

Pozice policistů a zloděje budeme zapisovat jako dvojice (a, b), kde a je číslo horizontální ulice a b číslo vertikální ulice, které se protínají na křižovatce, kde daná osoba stojí. Nechť $r = 3$ a $s = 4$, policisté začínají na pozicích $(2, 0)$ a $(0, 1)$ a zloděj na pozici $(1, 2)$. Počáteční situace tedy vypadá následovně:

	0	1	2	3
0		P		
1			Z	
2	P			

Honička může vypadat třeba takto:

1. Oba policisté se pohnou o jednu křižovatku na východ, tedy na pozice $(2, 1)$ a $(0, 2)$.
2. Zloděj má jedinou možnost, jak se vyhnout okamžitému chycení: pohne se na východ na pozici $(1, 3)$.
3. První policista se posune na východ na pozici $(2, 2)$, druhý na jih na pozici $(1, 2)$.
4. Zloděj by mohl ještě chvíli utíkat a pohnout se na sever, nicméně se rozhodl to dále neprotahovat a zůstat stát na pozici $(1, 3)$.
5. První policista zůstane na místě, druhý se pohne na východ na pozici $(1, 3)$ a chytí zloděje.

P-I-4 O Vekslákbotovi a Pokladničce

K této úloze se vztahuje studijní text uvedený na následujících stranách. Doporučujeme vám nejprve si přečíst studijní text a až potom se vrátit k samotným soutěžním úkolům. Ze stejného studijního textu budou vycházet i úlohy v dalších kolech soutěže.

Ve všech třech podúlohách je na začátku v Pokladničce $c > 0$ červených žetonů, $m > 0$ modrých žetonů a nic jiného. Jednotlivé podúlohy jsou nezávislé, například podúlohy b) a c) můžete řešit, i když nevyřešíte podúlohu a). Připomínáme, že při hodnocení této úlohy *nezáleží na časové složitosti* vašeho řešení, pouze na jeho správnosti a vašem popisu a zdůvodnění.

a) *(3 body)*

Napište program, po jehož skončení bude v Pokladničce opět $c+m$ žetonů, ale jiné barvy: Pokud na začátku bylo více červených než modrých, budou všechny žetony růžové, jinak budou všechny žetony tyrkysové.

b) *(3 body)*

Napište program, po jehož skončení bude v Pokladničce přesně c červených, m modrých a $c + m$ fialových žetonů. (Navíc v ní může být i libovolné množství žetonů jiných barev.) Upozorňujeme, že váš program musí skončit, tedy v koncovém stavu nesmí být možné vykonat žádnou z jeho instrukcí.

c) *(4 body)* Napište program, po jehož skončení bude v Pokladničce přesně $c \cdot m$ fialových žetonů a nic jiného.

Pokud vyrobíte přesně $c \cdot m$ fialových žetonů, ale ještě vám zůstanou i žetony dalších barev, dostanete 3 body. Pokud navíc před začátkem výpočtu v Pokladničce kromě červených a modrých žetonů potřebujete mít i vámi přesně určenou sadu žetonů dalších barev (jejichž počet nesmí záviset na c a m), dostanete 2 body.

Studijní text

Za devatero horami a devatero řekami byla jednou jedna prazvláštní země. V této zemi žili roboti a místo peněz používali žetony všech možných barev. V domečku na úpatí desáté hory spolu žili dva hrdinové našeho příběhu: roboti Vekslákbot a Pokladnička.

Jak asi tušíte z jejího jména, Pokladnička v sobě ráda ukládá všechny možné žetony. Jak možná z jejího jména netušíte, Pokladnička také velmi ráda vykonává různé programy. A jak jste si po přečtení předchozí věty naprosto jistí (ostatně, toto je studijní text Matematické Olympiády kat. P a ne jen tak nějaká pohádka), právě tyto programy pro ni budete psát vy jako řešení soutěžních úloh.

Vekslákbot je mistr vyměňování žetonů. Ať už chce vyměnit dva červené za tři modré, anebo žlutý a černý za $10^9 + 7$ šedých, Vekslákbot určitě zná robota, který zná robota, jenž s ním právě takovou směnu rád provede. Vekslákbot má moc rád Pokladničku, takže pokud ho ona o nějakou výměnu poprosí, okamžitě (nebo, jak říkáme my, v konstantním čase) ji pro ni provede.

Pojďme se nyní podívat na to, jak budou vypadat programy pro Pokladničku.

Základy: vstup, výstup, program

Vstupem pro Pokladničku budou žetony, jež má na začátku uložené v sobě. Neexistuje žádný výstup. V zadání jednotlivých úloh budeme různě definovat cíle, jichž mají vaše programy dosáhnout.

Program pro Pokladničku se skládá ze dvou částí: *omezení*, která musí dodržet, a *pokynů*, které má vykonávat.

Omezení

První částí programu pro Pokladničku je **konečná množina** omezení (může být i prázdná). Omezení pro Pokladničku určují, kolik nejvýše žetonů konkrétní barvy, případě kombinací žetonů různých barev, může mít najednou v sobě. Formálně, omezení jsou lineární nerovnosti následujícího tvaru:

$$k_1 \cdot \text{barva}_1 + k_2 \cdot \text{barva}_2 + \cdots + k_n \cdot \text{barva}_n \leq \text{limit},$$

kde všechny koeficienty k_i i limit jsou konkrétní kladná celá čísla a barva_i jsou proměnné označující počet žetonů příslušné barvy v Pokladničce. Tečky $(\cdot)$ můžeme vynechat, pokud je zřejmé, kde končí koeficient a kde začíná název barvy. Příklady omezení jsou například nerovnosti $\text{modrá} \leq 7$ nebo $\text{modrá} + 2\text{červená} \leq 3$.

Pokud se nějaké barvy žádné omezení netýká, může být v Pokladničce libovolné množství žetonů této barvy.

Instrukce

Každá instrukce pro Pokladničku má následující tvar:

$$k_1 \cdot \text{barva}_1, \ldots, k_n \cdot \text{barva}_n \to \ell_1 \cdot \text{barva}_{n+1}, \ldots, \ell_m \cdot \text{barva}_{n+m}$$

Všechno nalevo od $\to$ budeme nazývat levou stranou instrukce, všechno napravo zase pravou stranou. Tečky $(\cdot)$ můžeme vynechat, pokud je zřejmé, kde končí koeficient a kde začíná název barvy.

Všechna k_i i ℓ_j musí být kladná celá čísla. V rámci pravé i levé strany instrukce musí být všechny použité barvy různé. Je povolené použít stejnou barvu na obou stranách instrukce. Je povolené mít $n = 0$ či $m = 0$, tedy instrukci, které má některou stranu prázdnou. (Je povolené mít prázdné i obě strany, ale jak se brzy dozvíme, program, v němž bychom takovou instrukci použili, by nikdy neskončil.)

Příklady instrukcí:

- 2červená $\to$ 3modrá
- 1žlutá, 1černá $\to$ $(10^9 + 7)$šedá
- 3červená $\to$ 333červená, 334cyklámenová, 335purpurová
- 2zelená $\to \emptyset$

Poslední instrukce z příkladu má prázdnou pravou stranu. Pro zápis prázdné strany instrukce budeme používat symbol prázdné množiny, aby bylo jasné, že je prázdná úmyslně.

V našich příkladech budeme používat skutečné názvy barev. Ve svých programech můžete jako názvy barev používat i libovolné jiné alfanumerické řetězce. Je také povoleno v zápisech vynechávat koeficient 1. Druhou z výše uvedených instrukcí lze tedy zapsat také jako žlutá, černá $\to$ $(10^9 + 7)$šedá.

Pokladnička vykoná instrukci tak, že ze sebe vyndá sadu žetonů na levé straně instrukce, dá ji Vekslákbotovi a poprosí jej, aby jí za ně přinesl sadu žetonů z pravé strany. Vekslákbot to samozřejmě okamžitě zajistí a Pokladnička do sebe vloží žetony, které jí přinesl.

Pokud v sobě Pokladnička nemá všechny žetony, které vyžaduje levá strana instrukce, nelze danou instrukci v tu chvíli provést. Pokud by například měla pouze dva červené žetony, nešlo by provést instrukci 3červená $\to$ 333červená.

Pokladnička také nemůže provést instrukci, po jejímž provedení by bylo porušené některé omezení.

Program

Program pro Pokladničku tvoří **konečná posloupnost** instrukcí výše uvedeného typu. Zdůrazňujeme, že jde o posloupnost, tedy **záleží** na pořadí instrukcí.

Vykonávání programu

Program se vykonává v krocích. V každém kroku Pokladnička začne číst program od začátku a čte jej, dokud nenajde první instrukci, kterou momentálně může vykonat, a tu vykoná. (Na zbytek programu se v tomto kroku už ani nepodívá a v dalším kroku začne znovu číst program od začátku.)

Vykonávání programu skončí, když už žádnou instrukci v programu nelze vykonat.

Příklad #1: Více červených

Úloha: Na začátku jsou v Pokladničce nějaké červené a nějaké modré žetony. Napište program, po jehož skončení bude v Pokladničce právě jeden zlatý žeton a nic jiného, pokud bylo červených žetonů více než modrých. Ve všech ostatních případech musí Pokladnička skončit úplně prázdná.

Řešení 1 (bez omezení)

Nebudeme mít žádná omezení. Pokyny budou vypadat následovně:

1. červená, modrá $\rightarrow \emptyset$
2. červená, zlatá $\rightarrow$ zlatá
3. červená $\rightarrow$ zlatá
4. modrá $\rightarrow \emptyset$

Při vykonávání tohoto programu bude nejprve Pokladnička používat instrukci 1, dokud to lze. Až to přestane být možné, má v sobě už buď jen červené, anebo jen modré žetony (anebo žádné žetony a program skončí). Pokud jsou modré, jediná vykonatelná instrukce je instrukce 4, jejímž opakovaným používáním se Pokladnička vyprázdní a program skončí.

Pokud po skončení instrukce 1 zůstanou v Pokladničce červené žetony, bude to o trochu složitější: Jediná instrukce, která se v danou chvíli dá použít, je instrukce 3, která vymění jeden červený žeton za jeden zlatý. Od této chvíle se však instrukce 3 již nepoužije, a to proto, že lze vykonávat instrukci 2, která je v posloupnosti dříve. Pomocí té se Pokladnička postupně zbaví zbývajících červených žetonů. Jakmile v ní zůstane jen samotný zlatý žeton, nelze vykonat žádnou instrukci, a program tedy skončí.

Řešení 1 (s omezením)

Stejnou úlohu můžeme vyřešit s použitím omezení. Vystačíme si s jediným:

- zlatá ≤ 1

Program vypadá následovně:

1. červená, modrá $\rightarrow \emptyset$
2. červená $\rightarrow$ zlatá
3. červená $\rightarrow \emptyset$
4. modrá $\rightarrow \emptyset$

Tentokrát se v situaci, kdy jsou v Pokladničce jen samé červené žetony, nejprve jednou vykoná instrukce 2 (jeden červený žeton vyměníme za jeden zlatý) a potom se už bude používat instrukce 3, dokud červené žetony nedojdou. Omezení, které jsme si zvolili, totiž Pokladničce brání znovu využít instrukci 2.

Příklad #2: Součet

Úloha: Na začátku je v Pokladničce $c > 0$ červených žetonů, $m > 0$ modrých žetonů a jeden zelený. Napište program, po jehož skončení bude v Pokladničce přesně c červených, m modrých a $c + m$ fialových žetonů.

Řešení: Kdybychom pouze chtěli dostat $c + m$ fialových žetonů, stačilo by vyměnit všechny červené i modré žetony za fialové „s kurzem jedna ku jedné“. Jak ale nepřijít o původní žetony?

Naše řešení nebude mít žádná omezení a program bude vypadat následovně:

1. červená, zelená $\rightarrow$ tmavočervená, zelená
2. modrá, zelená $\rightarrow$ tmavomodrá, zelená
3. zelená $\rightarrow$ žlutá
4. tmavočervená, žlutá $\rightarrow$ červená, fialová, žlutá
5. tmavomodrá, žlutá $\rightarrow$ modrá, fialová, žlutá
6. žlutá $\rightarrow$ $\emptyset$

Všimněte si, jak náš program využívá přítomnost zeleného a žlutého žetonu v Pokladničce: Pomocí nich umíme rozlišit, zda ještě měníme původní červené a modré žetony, anebo zda již zpátky vznikají nové.