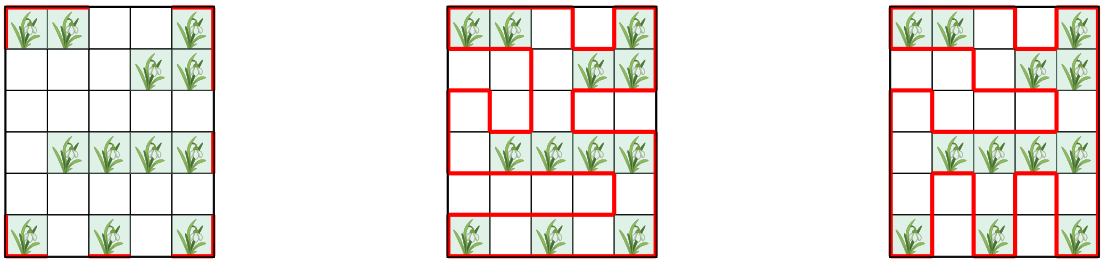
Kategorie mladší

Úloha 1A Elf Marek

Z celé ohraničené louky budeme „vykousávat“ jednotlivé čtverečky a tím zvyšovat obvod pastviny, viz obrázek 1. „Průchody“ mezi druhým a čtvrtým řádkem a čtvrtým a šestým řádkem lze „posouvat“ doleva a doprava a obvod zůstane stejný (vždy se dva metry odečtou na jedné straně a na druhé straně se přičtou). Takovýto plot má obvod 42 m. Nešlo by to ale lépe?

Z toho, že Markův plot se nikde nekříží a na plánku jde namalovat jednou lomenou čarou, vyplývá, že z každého vrcholu čtvercové sítě může plot vycházet maximálně do dvou směrů (ve skutečnosti musí buď vycházet právě do dvou směrů – to když plot daným bodem prochází, nebo jsou všechny hrany s bodem sousedící prázdné – to když plot bodem neprochází). V našem řešení platí, že plot prochází každým bodem čtvercové sítě, a proto si můžeme být jistí, že delší ohradu Marek opravdu postavit nemůže.



(a) (b) (c)

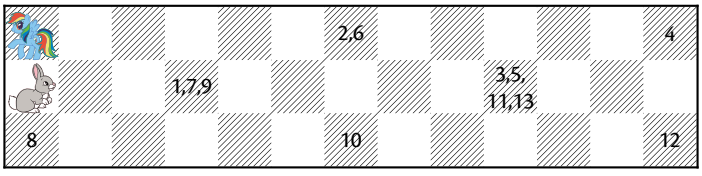
Obrázek 1

Úloha 2A Hyperkůň a králíček

Jak je patrné ze vzhledu závodní dráhy a pohybu hyperkoně, problematický bude každý lichý skok, kdy se hyperkůn dostane do dráhy Aprikotce. Sudé skoky naopak problém nepředstavují, jelikož v těchto skocích bude hyperkůn vždy ve své vlastní dráze.

Aprikotka se pohybuje vždy jen o jedno pole směrem k cíli, ve svém prvním tahu skočí z prvního na druhé pole, ve druhém tahu z druhého pole na třetí, atd. Co to znamená pro hyperkoně? Protože začíná, nesmí ve svém prvním tahu skončit na prvním poli (protože právě tam je Aprikotka) a ani nesmí skočit na pole druhé (protože tam skočí ve svém tahu Aprikotka). Jak jsme již napsali výše, druhý tah hyperkoně netrápí. Až ve třetím tahu opět musí skákat tak, aby neskončil na poli třetím, ani čtvrtém. A už je zde vidět nějaké pravidlo: Pro každý lichý tah k platí, že hyperkůň nesmí skočit na poli k ani $(k+1)$. Těchto skoků musíme udělat celkem třináct. S tímto omezením na paměti můžeme začít prozkoumávat možná řešení.

Velmi rychle se ukáže, že jedno z možných řešení je skákat směrem k cíli a těsně před cílovou páskou se otočit a zase skákat směrem zpět na start. Pak může hyperkůň zopakovat stejný postup, tentokrát ale po druhé straně závodní dráhy.



Obrázek 2

Na závěr stačí jen zkontrolovat jediná dvě políčka, která kolidují s dráhou Aprikotky – pole 4 a 10. Dle našeho zjištění víme, že na těchto políčkách se nesmí hyperkůň objevit v tahu 3, resp. 9. Tato čísla se tam ale nenacházejí, takže k žádnému zranění nedojde.

Úloha 3A Ananas

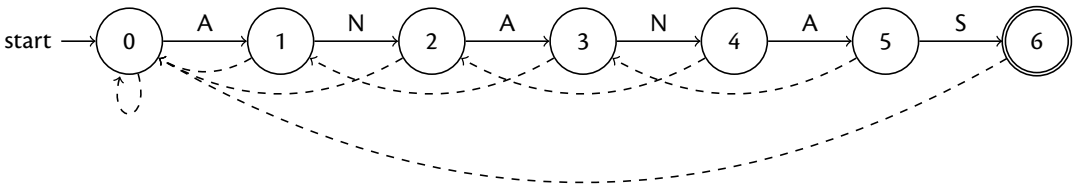
Dříve, než se pustíme do řešení problému, je dobré si ujasnit, proč vlastně robot velbloudům nefunguje tak, jak by měl. Vezměme si příklad ze zadání: text „anananas“, v němž pomocí robota hledáme výskyt slova „ananas“. Až do pátého písmenka je všechno v pořádku – robot přečetl „anana“ a nachází se nyní ve stavu 5. Pokud by nyní přečetl písmenko „s“, přesunul by se do stavu 6 a ohlásil by výskyt. To se ale nestane, protože na vstupu robot dostane další „n“. Přesune se tedy zpátky na začátek (do stavu 0) a pokračuje ve čtení písmenky

„a“ (přesune se do stavu 1) a „s“ (vrátí se zpět do stavu 0). Robot se tedy nechal zlákat slibně vypadajícím začátkem textu a nevšiml si, že kdyby začal se čtením o dvě písmenka dále, „ananas“ by našel.

Jedno možné řešení je tak nasnadě: Začneme číst na začátku textu, přečteme šest písmenek (protože slovo „ananas“ má právě šest písmen), a pokud se nyní budeme nacházet ve stavu 6, ohlásíme výskyt. Nyní se bez ohledu na to, ve kterém stavu jsme skončili, přesuneme zpět do stavu 0, v textu se vrátíme o pět písmenek zpátky (tedy k písmenku, které jsme předtím přečetli jako druhé) a pokus budeme opakovat. Pro náš vzorový vstup „anananas“ by tedy tento postup fungoval následovně: V prvním kole přečteme písmenka „ananan“, skončíme ve stavu 0, „ananas“ nenalezen. Nyní se vrátíme zpět ke druhému písmenku textu (do stavu 0 se vracet nemusíme, neboť v něm jsme), přečteme „nanana“, skončíme ve stavu 1, „ananas“ jsme opět nenalezli. Vrátíme se do stavu 0, začátek slova opět o jedna posuneme (tedy na druhé písmenko „a“), přečteme „ananas“ a hurá, jsme ve stavu 6, výskyt nalezen!

Toto řešení nám všechny výskyty slova „ananas“ v textu určitě najde, sám/sama ale asi vidíš, že není úplně efektivní. Každé písmenko textu při něm totiž musíme přečíst šestkrát! Časová náročnost by se dala trošku vylepšit tím, že bychom řekli, že pokud v právě čtené šestici narazíme na chybu, víme, že už určitě nebude správná, a můžeme začátek slova rovnou o jedno písmenko posunout. V našem příkladě by nám to v prvním kole (při čtení „ananan“) moc nepomohlo, tak jako tak bychom museli přečíst všech šest písmenek, v kole druhém bychom ale hned u prvního písmenka („n“) zjistili, že není správné, a rovnou bychom pokračovali kolem třetím, tedy čtením „ananas“.

Existuje ale i lepší řešení. Proč číst jedno a totéž písmenko několikrát? Nedalo by se raději nějakým chytrým způsobem upravit přerušované šipky? V našem příkladě je problém v tom, že poté, co robot přečte chybné písmenko (třetí „n“), přesune se zpátky do stavu 0 i přesto, že kdyby se čtením slova začal o dvě písmenka vedle, nacházel by se teď ve stavu 4. Jde nám tedy o to, aby se robot v případě, že narazí na chybu, přesunul do nejvzdálenějšího (rozuměj s největším číslem) stavu takového, že by se v něm mohl nacházet, kdyby se čtením začal o několik písmenek dál. Jinými slovy, hledáme co nejdelší konec dosud přečteného slova (budeme mu říkat přípona) takový, že je zároveň začátkem (neboli předponou) slova „ananas“. Třeba pro slovo „anana“ je takovou příponou slovo „ana“ a přerušovaná šipka ze stavu 5 nepovede do stavu 0, ale do stavu 3. Podobně přepojíme i ostatní přerušované šipky: Pro slova „a“ a „an“ žádná vhodná přípona neexistuje, šipky tedy povedou stále do stavu 0. Ze slova „ana“ můžeme odpojit „a“, a přesuneme se tedy do stavu 1, pro slovo „anan“ máme příponu „an“ a stav 2. Slovo „anana“ jsme již vyřešili, pro „ananas“ žádná vhodná přípona neexistuje a šipka tedy stále povede do stavu 0. Robot nyní vypadá takto:

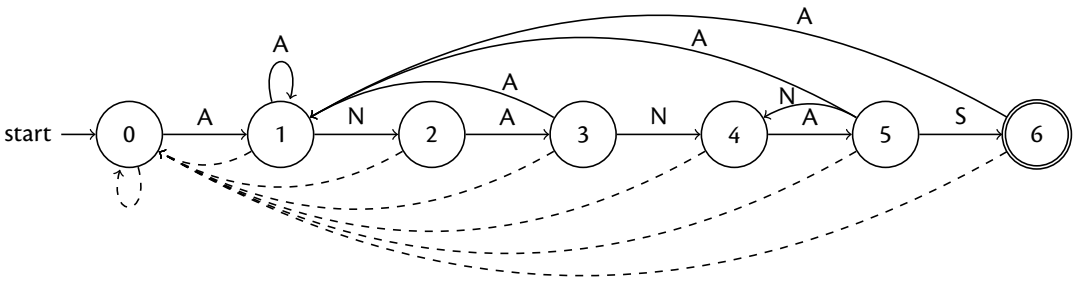


Obrázek 3

Jak bude robot fungovat? Dokud bude číst správná písmenka, bude po šipkách postupovat vpřed. Ve chvíli, kdy přijde nějaké jiné písmenko, než je nyní na šipce dopředu, přesune se robot po přerušované šipce zpět a podívá se, zda z tohoto stavu nevede šipka označená písmenkem na vstupu. Pokud ano, přesune se po ní dopředu a bude ve čtení pokračovat, pokud ne, přesune se opět po přerušované šipce zpátky, a takto bude pokračovat, dokud se nedostane do stavu 0. Pokud ani z tohoto stavu nepovede šipka označená daným písmenkem, zůstane robot ve stavu 0 a přečte z textu další písmenko.

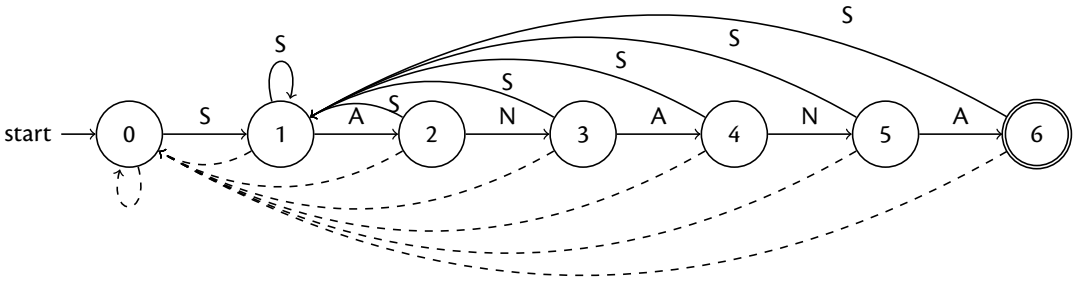
V našem příkladě tak robot přečte „anana“, nachází se ve stavu 5 a očekává písmenko „s“, na vstupu je ale „n“. Robot se tedy po přerušované šipce vrátí do stavu 3 a podívá se, jaké písmenko je na šipce, která z tohoto stavu vede. Je to „n“, robot se po ní tedy přesune do stavu 4 a po přečtení písmenek „a“ a „s“ zcela správně ohlásí výskyt.

S pomocí stejné myšlenky můžeme robota upravit i jinak, a to tak, že všechny čárkované šipky necháme tak, jak jsou, přidáme ale některé plné šipky, označené písmenky, vedoucí o několik stavů zpět. Třeba ze stavu 5 („anana“) bychom chtěli vést šipku označenou písmenkem „n“ do stavu 4 (odpovídá to našemu předchozímu postupu, kdy jsme po přečtení písmenka „n“ ve stavu 5 nejprve skočili do stavu 3 a poté se po šipce „n“ přesunuli dopředu). Žádné jiné písmenko nám po přidání k „anana“ nějakou správnou příponu nevytvoří, a po jejich přečtení se tedy můžeme, stejně jako předtím, přesunout rovnou do stavu 0. Povedou nějaké nové šipky i z ostatních stavů? Po „n“ následuje ve slově „ananas“ vždy písmenko „a“, a ze stavů 2 a 4 tak žádné nové šipky nepovedou. Ve stavu 1 musíme přidat šipku pro písmenko „a“ – mohlo by to být první písmenko „ananasu“, a proto místo vracení do stavu 0 chceme zůstat ve stavu 1. Ze stejného důvodu se do stavu 1 chceme vrátit rovněž v případě, že nám písmenko „a“ přijde na vstup ve stavu 3, 5 nebo 6. Upravený robot nyní vypadá takto:



Obrázek 4

Zajímavý nápad měli někteří z vás: Všimli si, že zatímco písmenka „a“ a „n“ se ve slově „ananas“ opakují, písmenko „s“ je unikátní. Pokud bychom tedy místo začátku „ananasu“ hledali konec, mohl by být robot jednodušší. V zásadě stačí číst vstupní text pozpátku a šipky v robotovi přepojit opačným směrem, nesmíme ale zapomenout na jednou maličkost, a to, že v textu se může vyskytovat více písmenek „s“ za sebou. Buď tedy musíme robotovi opět říct, že pokud se mu nepovede popojít ze současného stavu vpřed, má se vrátit po přerušované šipce a pokusit se udělat krok vpřed s tím samým písmenkem, nebo přidáme šipku označenou písmenkem „s“ ze stavu 6 do stavu 6:



Obrázek 5

Úloha 4A Náramky pro lišky

Abychom zjistili délku klacíků liščí paní učitelky, můžeme se na problém podívat od konce. Jak se získá obvod pro liščího cvalíka s 40 cm obvodem zápěstí je jasné – klacíky se prostě vyskládají za sebe. Pro získání 39 cm je pak jediná možnost a to je odebrání 1 cm, což bude také délka prvního klacíku. Potom není problém ani 38 cm – onen první klacík se přiloží k ostatním a tak lze jeho délku od 39 cm odečíst. Pro 37 cm už ale budeme potřebovat další klacík, tentokrát o délce 3 cm. Díky těmto dvěma klacíkům dovedeme bez potíží poskládat délky od 36 cm až po 32 cm (u níž od 40 dvakrát odečteme 1 a 3, jednou pro odebrání klacíku a jednou pro odečtení hodnoty s jeho pomocí). Pro 31 cm budeme potřebovat třetí klacík, tentokrát o délce 9 cm. Poslední klacík pak nutně musí být takový, aby v součtu s ostatními dal oněch počátečních 40 cm, zde se tedy bude jednat o délku 27 cm. Máme tedy již hledané čtyři délky větviček, a pokud chceme být obzvláště pečliví, můžeme projít všechny hodnoty od 1 cm do 40 cm a najít kombinace poskládání těchto čtyř klacíků, které by jim odpovídalo, což je skutečně možné.

Známe tedy délku větviček, které paní učitelka našla – jejich rozměry jsou 1 cm, 3 cm, 9 cm a 27 cm.

Úloha 5A Loď

Ze všeho nejdřív musí Eda určitě nalodit sám sebe, zbývající náklad tak může čítat už jenom 12 tun. Do těchto 12 tun se musí vejít nejen jeho přátelé, ale samozřejmě i tolik důležité dárky. Na každého velblouda tedy bude muset obětovat 620 kg z nosnosti lodi a na každého slona 1 035 kg. Nyní by si měl Eda spočítat, kdy se mu vyplatí vzít spíše slony a kdy spíše velbloudy. Stejné množství dárků může získat buď od 4 slonů, nebo od 7 velbloudů (v obou případech to bude 140 kilogramů kořínků), s tím, že kapacita místa na lodi v případě slonů klesne o 4 140 kilogramů, u velbloudů o 4 340 kilogramů. Edovi se tedy určitě vyplatí pozvat raději 8 slonů než 14 velbloudů a kapacita lodi se tak sníží na pouhých 3 720 kg. Takovou hmotnost má přesně 6 velbloudů i s dárky nebo 3 sloni s tím, že nějaké místo ještě zbude (konkrétně zůstane nevyužito 615 kg, což je tak akorát, aby se nám už nevešel ani slon, ani velbloud). Eda samozřejmě může pozvat také jednoho slona (a zbytek doplnit velbloudy), nebo dva slony (a zbytek opět doplnit velbloudy). Co bude nejvýhodnější? Šest velbloudů donese 120 kg dobrot. K jednomu slonovi může Eda přibrat 4 velbloudy, dohromady od nich dostane 115 kg lotosových kořínků. Ke dvěma slonům se vejdou ještě dva velbloudi a tato skupinka by Edovi dohromady darovala 110 kg kořínků. A tři sloni, ti donesou „jenom“ 105 kg dobrot. Edovi se tedy vyplatí pozvat skupinku osmi slonů a šesti velbloudů, od nichž získá krásných 400 kg lotosových kořínků. Dobrou chuť a šťastnou plavbu!

Kategorie starší

Úloha 1B Domino

Každá kostička domina se skládá z 1 bílého a z 1 černého políčka. Nezbeda myška nám ale vykousla dvě bílá políčka, a tak nám zbývá 30 bílých a 32 černých políček. To znamená, že nám vždycky zůstanou 2 černá pole, ze kterých domino nevytvoříme – slepování je zakázané.

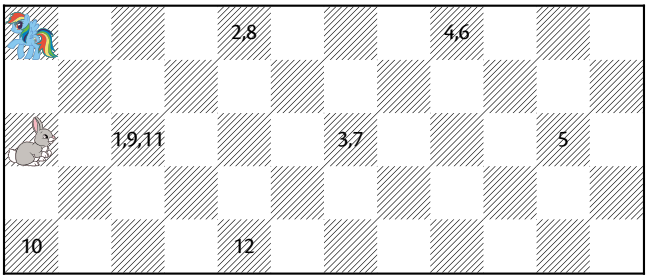
Odpověď tedy zní: Brumlovi se nepovede vytvořit všech 31 kostiček domina.

Úloha 2B Hyperkůň a králíček

Jak je patrné ze vzhledu závodní dráhy a pohybu hyperkoně, problematický bude každý lichý skok, kdy se hyperkůn dostane do dráhy Aprikotce. Sudé skoky naopak problém nepředstavují, jelikož v těchto skocích bude hyperkůn vždy ve své vlastní dráze.

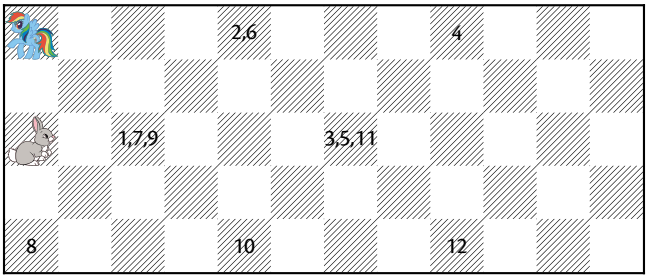
Aprikotka se pohybuje vždy jen o jedno pole směrem k cíli, ve svém prvním tahu skočí z prvního na druhé pole, ve druhém tahu z druhého pole na třetí, atd. Co to znamená pro hyperkoně? Protože začíná, nesmí ve svém prvním tahu skončit na prvním poli (protože právě tam je Aprikotka) a ani nesmí skočit na pole druhé (protože tam skočí ve svém tahu Aprikotka). Jak jsme již napsali výše, druhý tah hyperkoně netrápí. Až ve třetím tahu opět musí skákat tak, aby neskončil na poli třetím, ani čtvrtém. A už je zde vidět nějaké pravidlo: Pro každý lichý tah k platí, že hyperkůň nesmí skočit na poli k ani $(k+1)$. Těchto skoků musíme udělat celkem dvanáct. S tímto omezením na paměti můžeme začít prozkoumávat možná řešení.

Nejspíš nás jako první napadne jednoduché řešení skákat směrem k cíli a těsně před cílovou páskou se otočit a zase skákat směrem zpět na start. Pak může hyperkůň zopakovat stejný postup, tentokrát ale po druhé straně závodní dráhy.



Obrázek 6

Zkontrolujeme jediná tři políčka, která kolidují s dráhou Aprikotky – pole 3, 7 a 11. Dle našeho zjištění víme, že na těchto políčkách se nesmí hyperkůň objevit v tahu 3, resp. 7 a 11. Bohužel ale zjistíme, že v tomto případě by na sedmém poli došlo ke zranění, musíme tedy hyperkoně otočit dříve, aby jeho sedmý skok dopadl na jiné pole.



Obrázek 7

Tímto způsobem dokonce zmenšíme šanci, že se hyperkůň s Aprikotkou potkají a musíme zkontrolovat pouze pole 3 a 7. Na žádném z nich se nenachází problém, takže teď již k žádnému zranění nedojde.

Úloha 3B Marticové počty 3

Ze všeho nejdřív se hodí zjistit, jaké musí mít martice R a S rozměry. Nápověda pod zadáním nás nabádá, abychom nezapomněli, že marticové násobení není komutativní, proto budeme raději předpokládat, že máme dvě takové martice R, jednu pro násobení zleva a druhou pro násobení zprava (budeme je značit R_L a R_P), a stejně tak také dvě martice S (opět budeme značit S_L a S_P). O marticích R_L, R_P, S_L a S_P nyní víme:

$$R_L \cdot M = M \cdot R_P = M_{1*} \quad \text{a} \quad S_L \cdot M = M \cdot S_P = M_{*1},$$

kde M_{1*} a M_{*1} značíme první řádek, resp. první sloupec martice M. Z úlohy 4B z prvního kola soutěže víme, že aby spolu šly dvě martice vynásobit, musí mít první martice tolik sloupců, kolik má druhá martice řádků, a že výsledná martice „zdědí“ počet řádků od první a počet sloupců od druhé martice (matematičtěji řečeno, násobit spolu můžeme martice o rozměrech $p \times q$ a $q \times r$ a výsledná martice má rozměry $p \times r$; rozměry jsou vždy ve tvaru počet řádků $\times$ počet sloupců). Rozměry martice M budeme obecně značit $m \times n$, výsledné martice pak mají rozměry $1 \times n$ (M_{1*}) a $m \times 1$ (M_{*1}). Z toho už snadno odvodíme, že R_L musí mít rozměry $1 \times m$ a S_P $n \times 1$, a také to, že martice R_P a S_L obecně existovat nebudou. (Například pro násobení marticí R_P by totiž o rozměrech martic muselo platit: $m \times n \cdot n \times ? \to m \times ?$, my ale zároveň víme, že výsledná martice musí mít rozměry $1 \times n$. Martice R_P tedy existuje jen pro martice, které mají jen jeden řádek, a výsledkem po vynásobení bude tentýž jeden řádek, tedy přímo martice M. Martice R_P tedy bude marticová jednička (viz úloha 4B z prvního kola). Pro S_L provedeme úvahu úplně stejně.)

Správné rozměry martic známe, zbývá zjistit, jak budou R_L a S_P vypadat. Z definice marticového násobení platí:

$$\begin{array}{c|c|c|c} r_1 & r_2 & \dots & r_m \end{array} \cdot \begin{array}{|c|c|c|c|} \hline m_{11} & m_{12} & \dots & m_{1n} \\ \hline m_{21} & m_{22} & \dots & m_{2n} \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline m_{m1} & m_{m2} & \dots & m_{mn} \\ \hline \end{array} =$$

$$\begin{array}{c|c|c|c} r_1 \cdot m_{11} + r_2 \cdot m_{21} + \dots + r_m \cdot m_{m1} & r_1 \cdot m_{12} + r_2 \cdot m_{22} + \dots + r_m \cdot m_{m2} & \dots & r_1 \cdot m_{1n} + r_2 \cdot m_{2n} + \dots + r_m \cdot m_{mn} \end{array}$$

a my chceme, aby se výsledek rovnal prvnímu řádku martice M, tedy

$$\begin{array}{c|c|c|c} m_{11} & m_{12} & \dots & m_{1n} \end{array}.$$

Zapsáno trošku přátelštějším způsobem pro konkrétní martici $M = \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}$ a $R_L = \begin{array}{c|c} r_1 & r_2 \end{array}$:

$$R_L \cdot M = \begin{array}{c|c} r_1 & r_2 \end{array} \cdot \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array} = \begin{array}{c|c} 1 \cdot r_1 + 3 \cdot r_2 & 2 \cdot r_1 + 4 \cdot r_2 \end{array} = \begin{array}{c|c} 1 & 2 \end{array}.$$

Z toho již snadno odvodíme, že pokud se r_1 bude rovnat 1 a všechny ostatní členy v martici R_L 0, dostaneme vždy požadovaný výsledek. Martice R_L tak má rozměry $1 \times m$ a tvar $\begin{array}{c|c|c|c} 1 & 0 & \dots & 0 \end{array}$.

Pro martici S_P pak stejnou úvahou dospějeme k tomu, že bude vypadat takto:

$$\begin{array}{|c|} \hline 1 \\ \hline 0 \\ \hline \vdots \\ \hline 0 \\ \hline \end{array}$$

Úloha 4B Loď

Z Archimédova zákona víme, že:

$$\begin{aligned} F_g &= F_{vz} \\ m \cdot g &= \rho \cdot V \cdot g \\ m_L + m_Z &= \rho_{H_2O} \cdot a \cdot b \cdot (c - h) \\ m_Z &= \rho_{H_2O} \cdot a \cdot b \cdot (c - h) - \rho_D \cdot d \cdot [a \cdot b + 2 \cdot b \cdot (c - d) + 2 \cdot (a - 2 \cdot d) \cdot (c - d)] \end{aligned}$$

F_g je síla tíhová, F_{vz} síla vztlaková, m_L je hmotnost lodi, m_Z je hledaná volná hmotnost (hmotnost zvířat, které si lišák Lukáš může pozvat), a, b, c jsou rozměry lodi (délka, šířka, výška), h je výška okraje lodi na hladinou, d je tloušťka prken, z nichž je loď postavená, ρ_{H_2O} je hustota vody, ρ_D je hustota dřeva. Při výpočtu hmotnosti lodi nesmím zapomenout započítat „překryvy" stěn lodi.

Když se dosadí příslušné hodnoty, hledaná volná hmotnost vyjde 39 503 921,36 kg. Součet hmotností Lukáše a všech jeho kamarádů je 11 064,37 kg, takže lišák Lukáš si může pozvat všechny své kamarády a ještě mu nějaká „volná" hmotnost zbude.

Úloha se dá řešit i jinými způsoby, tento je však nejkratší a fyzikálně i matematicky nejčistší.

Úloha 5B Velikonoční vajíčka

Abychom zjistili, jaký objem sádry budeme potřebovat na výrobu jedné originální velikonoční kuličky, musíme odečíst objem menší kuličky od té větší. Proto si nejprve spočítáme jejich objemy. Nesmíme zapomenout na to, že v zadání máme zadané průměry, zatímco pro vypočtení objemu potřebujeme poloměr (neboli polovinu průměru). Zároveň si poloměry rovnou převedeme na decimetry, jelikož objem sádry máme zadaný v litrech a 1 l = 1 dm^3

Objem větší kuličky:

$$V_1 = \frac{4}{3}\pi r_1^3$$
$$V_1 = \frac{4}{3}\pi \cdot 0{,}4^3 \doteq 0{,}268\,\mathrm{dm}^3 = 0{,}268\,\mathrm{l}$$

Objem menší kuličky:

$$V_2 = \frac{4}{3}\pi r_2^3$$
$$V_2 = \frac{4}{3}\pi \cdot 0{,}2^3 \doteq 0{,}034\,\mathrm{dm}^3 = 0{,}034\,\mathrm{l}$$

Objem velikonoční kuličky:

$$V = V_1 - V_2$$
$$V = 0{,}268 - 0{,}034 = 0{,}234\,\mathrm{l}$$

K dispozici máme 2,5 l sádry. Počet kamarádů, které může Tomáš kuličkami podarovat, spočítáme jako podíl celkového objemu sádry a 1 velikonoční kuličky.

$$n = \frac{2{,}5}{0{,}234} \doteq 10{,}684$$

2,5 litrů sádry nám postačí na 10,684 kuličky. Vzhledem k tomu, že by se asi nikdo nespokojil s 0,684 kuličky, může Tomáš vyrobit 10 kuliček.