Pavel Pokorný

Přelévání vody mezi nádobami, teorie grafů a modulární aritmetika

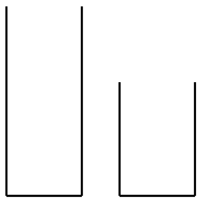
Rozhledy matematicko-fyzikální, Vol. 98 (2023), No. 1, 9–24

Persistent URL: http://dml.cz/dmlcz/151593

Přelévání vody mezi nádobami, teorie grafů a modulární aritmetika

Pavel Pokorný, VŠCHT Praha

Uvažujme dvě nádoby na vodu, jednu o objemu 5 litrů, druhou o objemu 3 litry, ale bez rysek udávajících výšku hladiny (viz obr. 1). Můžeme jednu nebo druhou nádobu zcela naplnit čerstvou vodou, můžeme libovolnou nádobu zcela vyprázdnit (tím, že vodu vylijeme do odpadu) nebo můžeme vodu z jedné nádoby přelít do druhé nádoby, a to tak, že buďto přelijeme všechnu vodu, pokud se do druhé nádoby vejde, tedy až do stavu, kdy je první nádoba prázdná, nebo můžeme přelít takové množství vody, až je druhá nádoba zcela plná.



Obr. 1: Schématický náčrtek velké nádoby o objemu 5 litrů (vlevo) a malé nádoby o objemu 3 litry (vpravo)

Úloha zní: lze s původně prázdnými nádobami dosáhnout stavu, kdy je v jedné nádobě 1 litr vody?

Uvedeme rychlé intuitivní řešení této úlohy a pak si ukážeme některé zajímavé matematické pojmy a algoritmy, které jsou užitečné i v jiných souvislostech.

Intuitivní řešení

Když zkoušíme téměř náhodně, jaké jsou dovolené operace, tak po chvíli nejspíše dospějeme k této možnosti: Naplníme malou nádobu po okraj, tedy aby v ní byly 3 litry vody. Pak tuto vodu přelijeme do velké nádoby. Potom opět malou nádobu naplníme po okraj a nakonec vodou z malé nádoby doplníme velkou nádobu, aby byla plná, tedy přelijeme 2 litry. Tak zůstane v malé nádobě právě 1 litr vody.

A co kdybychom začali naopak tím, že naplníme na začátku velkou nádobu po okraj, tedy aby v ní bylo 5 litrů? Jak lze dále postupovat? Můžeme přelít 3 litry do malé nádoby a ve velké nádobě zůstanou 2 litry. Pak malou nádobu vyprázdníme. Poté 2 litry z velké nádoby přelijeme do malé a velkou opět naplníme po okraj. Nyní můžeme malou nádobu doplnit z velké tím, že z ní přelijeme 1 litr, takže ve velké budou 4 litry a malá bude plná. Pak malou nádobu vyprázdníme. Nyní z velké nádoby, ve které jsou 4 litry, můžeme přelít do malé nádoby 3 litry a ve velké nádobě zbude 1 litr. Tento druhý způsob je zjevně delší. Nabízí se otázka, jak najít nejkratší postup vedoucí k danému cíli. A jaké stavy lze dosáhnout a jaké nelze?

V dalších kapitolách si ukážeme některé zajímavé matematické nástroje, které jsou užitečné pro hledání odpovědí na tyto a další otázky.

Matice přechodů

Pro pečlivý popis zavedeme veličiny: objem velké nádoby $a = 5$, objem malé nádoby $b = 3$, množství vody ve velké nádobě $x \in \{0, 1, \ldots, a\}$ a množství vody v malé nádobě $y \in \{0, 1, \ldots, b\}$. Pak celkový počet možných stavů je $n = (a+1)(b+1) = 24$. Očíslujme tyto stavy indexem $i = (b+1)x+y$; $i \in \{0, 1, 2, \ldots, n-1\}$. Tedy např. stav $(0; 0)$, kdy $x = 0$, $y = 0$ označíme indexem $i = 0$, stav $(0; 1)$, kdy $x = 0$, $y = 1$ označíme indexem $i = 1$, až stav $(5; 3)$, kdy $x = 5$, $y = 3$ označíme indexem $i = 23$. Zpětný výpočet, tedy jak nalézt stav $(x; y)$ z indexu i, lze provést takto: $y = i \bmod (b + 1)$, $x = (i - y)/(b + 1)$, kde mod je operace modulo, tedy zbytek po dělení. Např. pro stav $(2; 3)$, kdy $x = 2$, $y = 3$ je index $i = 4x + y = 11$ a naopak $y = 11 \bmod 4 = 3$ a $x = (11 - 3)/4 = 2$, protože 11 děleno 4 jsou 2 a zbytek je 3.

K operaci modulo se ještě vrátíme. Poznamenejme jen, že např. mod 10 vlastně dá z čísla zapsaného v desítkové soustavě číslici na místě jednotek, např. $158 \bmod 10 = 8$.

Nyní můžeme zavést matici přechodu (pozor, že v jiných souvislostech tento výraz může znamenat něco zcela jiného), tedy čtvercovou tabulku čísel, která má 24 řádků a 24 sloupců. Na řádku i ve sloupci j bude 0, jestliže nelze jedním krokem přejít ze stavu i do stavu j, a bude tam 1, jestliže to lze. Krokem zde rozumíme jednu dovolenou operaci s nádobami, tedy

- naplnění jedné nebo druhé nádoby
- vyprázdnění jedné nebo druhé nádoby

- přelití vody z jedné nádoby do druhé (a to tak, až bude buďto cílová nádoba plná, nebo až bude zdrojová nádoba prázdná)

Tak dostaneme následující matici

$$
\begin{pmatrix}
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
1&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1\\
1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&1&1&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&1&1&1&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1\\
1&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0\\
0&1&0&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&1&0&0\\
0&0&1&0&0&0&0&1&1&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1\\
1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0\\
0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&1&1&0&0&0&0&1&0&0\\
0&0&1&0&0&0&0&0&0&0&0&1&1&0&0&1&0&0&0&0&1&0&1&0\\
0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&1\\
1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1&1&1&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&0&1&1&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&1\\
1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&1\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0
\end{pmatrix}.
$$

Při ručním výpočtu je nebezpečí chyby, která se špatně odhaluje, proto je vhodné tyto úkony provést na počítači např. pomocí těchto příkazů v systému *Mathematica*:

```
a = 5;  (* velikost leve  nadoby *)
b = 3;  (* velikost prave nadoby *)
n = (a + 1)*(b + 1);    (* pocet stavu *)
t = Table[0, {n}, {n}]; (* pripravime si matici nul *)
s[x_, y_] := 1 + (b + 1) x + y; (* jak spocitat index *)
Table[
 t[[s[x, y], s[0, y]]] = 1; (* levou  nadobu lze vyprazdnit *)
 t[[s[x, y], s[x, 0]]] = 1; (* pravou nadobu lze vyprazdnit *)
 t[[s[x, y], s[a, y]]] = 1; (* levou  nadobu lze naplnit    *)
 t[[s[x, y], s[x, b]]] = 1; (* pravou nadobu lze naplnit    *)
```

```
 If[x > 0 && y < b && x <= b - y, t[[s[x, y], s[0,x+y]]]=1];
    (* je-li vlevo trochu vody, lze ji prelit doprava *)
 If[x > 0 && y < b && x >  b - y, t[[s[x, y], s[x+y-b,b]]]=1];
    (* je-li vlevo hodne vody, lze trochu ulit doprava *)
 If[x < a && y > 0 && y <= a - x, t[[s[x, y], s[x+y,0]]]=1];
    (* je-li vpravo trochu vody, lze ji prelit doleva *)
 If[x < a && y > 0 && y > a - x, t[[s[x, y], s[a,x+y-a]]]=1];
    (* je-li vpravo hodne vody, lze trochu ulit doleva *)
 t[[s[x, y], s[x, y]]] = 0,
    (* prechod do stejneho stavu je zbytecne uvazovat *)
 {x, 0, a}, {y, 0, b}];
TeXForm[t] (* vystup matice t ve formatu TeX *)
```

Např. na prvním řádku vidíme pouze dvě jedničky. Ve čtvrtém sloupci, tedy pro index $i = 3$, to odpovídá přechodu $(0; 0) \to (0; 3)$, kdy naplníme pravou nádobu, a v dvacátém prvním sloupci, tedy pro index $i = 20$, to odpovídá přechodu $(0; 0) \to (5; 0)$, kdy naplníme levou nádobu.

Dále vidíme, že některé sloupce obsahují pouze nuly, např. šestý sloupec, tedy s indexem $i = 5$ odpovídající stavu $(1; 1)$. To znamená, že do tohoto stavu se nelze dostat dovolenou operací z jiného stavu. Tedy pro naše další úvahy lze vynechat tento stav, tedy škrtnout sloupec a řádek s tímto indexem. Podobně odstraníme i další sloupce a řádky s indexem odpovídajícím nulovému sloupci (je jich celkem 8). Tak dostaneme následující matici přechodu mezi dosažitelnými stavy, kterou označíme P. Matice P má velikost 16×16, zatímco původní matice měla velikost 24×24.

$$P = \begin{pmatrix}
0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0\\
1&0&0&1&1&0&0&0&0&0&0&0&0&1&0&0\\
1&0&0&1&0&0&1&0&0&0&0&0&0&0&1&0\\
1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1\\
1&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&0&1&1&0&0&0&0&0&1&0&0&0&0&1\\
1&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0\\
0&0&0&1&0&0&1&0&0&0&0&0&1&0&0&1\\
1&0&0&1&0&0&0&0&0&1&0&0&1&0&0&0\\
0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&1\\
1&0&0&0&0&1&0&0&0&0&0&1&1&0&0&0\\
0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&1\\
1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1\\
0&1&0&0&0&0&0&0&0&1&0&0&1&0&0&1\\
0&0&1&0&0&0&0&0&0&0&0&1&1&0&0&1\\
0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0
\end{pmatrix}$$

Zde již je význam indexu jiný než v předchozí matici 24×24. Označme m počet přípustných stavů. Zde tedy $m = 16$. Stavy budeme opět číslovat od nuly.

Označme p_{ij} prvek matice P na řádku i ve sloupci j. Hodnota $p_{ij} = 1$ znamená, že se lze v jednom kroku dostat ze stavu i do stavu j. V opačném případě je $p_{ij} = 0$.

A co když budeme uvažovat dva kroky? Tedy v prvním kroku cestu ze stavu i do stavu j a ve druhém kroku cestu ze stavu j do stavu k. To bude možné, pokud $p_{ij} = 1$ a $p_{jk} = 1$, tedy pokud $p_{ij}p_{jk} = 1$ pro nějaký mezistav, tedy pro nějaké j. Takových přípustných mezistavů j může existovat i více. Pak součet součinů $\sum_{j=0}^{m-1} p_{ij}p_{jk}$ bude udávat počet cest ze stavu i do stavu k. Můžeme tedy zavést matici P^2 s prvky

$$p_{ik}^{(2)} = \sum_{j=0}^{m-1} p_{ij}p_{jk},$$

jejíž prvky $p_{ik}^{(2)}$ udávají počet cest ze stavu i do stavu k ve dvou krocích (zde dvojka v závorce je pouze pravý horní index, neznamená druhou mocninu čísla p_{ik} ani druhou derivaci, pochází z dvojky v symbolu P^2, který ale již má význam druhé mocniny matice P, jak ukážeme dále).

$$P^2 = \begin{pmatrix}
2&0&0&0&0&0&0&1&1&0&0&0&0&0&0&2\\
2&2&0&1&0&1&0&0&1&1&0&0&3&0&0&2\\
2&0&2&1&0&0&0&1&1&0&0&1&3&0&0&2\\
1&0&0&3&0&0&0&0&0&1&0&0&3&0&0&0\\
2&0&0&3&2&0&0&1&0&0&1&0&1&1&0&2\\
3&1&0&1&0&2&0&0&1&0&0&1&3&0&0&1\\
2&0&0&3&0&0&2&1&0&0&0&0&2&0&1&2\\
3&0&1&1&0&0&0&2&1&0&0&0&2&0&0&2\\
2&0&0&2&0&0&0&1&2&0&0&0&1&1&0&3\\
2&1&0&2&0&0&0&0&1&2&0&0&3&0&0&2\\
1&0&0&3&1&0&0&1&0&0&2&0&1&0&1&3\\
2&0&1&1&0&1&0&0&1&0&0&2&3&0&0&2\\
0&0&0&3&0&0&1&0&0&0&0&0&3&0&0&1\\
2&0&0&3&1&0&0&1&1&0&0&0&1&2&0&2\\
2&0&0&3&0&0&1&1&0&0&1&0&1&0&2&2\\
2&0&0&0&0&0&0&1&1&0&0&0&0&0&0&2
\end{pmatrix}$$

Tato operace se nazývá násobení matic a pro dvě matice A a B, kde matice A je typu $u \times v$, tedy má u řádků a v sloupců, a matice B je typu $v \times w$, tedy má v řádků a w sloupců, můžeme uvažovat jejich součin, což bude matice $C = A \cdot B$ s prvky

$$c_{ik} = \sum_{j=1}^{v} a_{ij} b_{jk},$$

což je vlastně skalární součin i-tého řádku matice A s k-tým sloupcem matice B. Je tedy nutné, aby šířka matice A, tedy počet jejích sloupců, se rovnal výšce matice B, tedy počtu jejích řádků.

Je-li význam prvku a_{ij} matice A počet cest ze stavu i do stavu j a význam prvku b_{jk} matice B počet cest ze stavu j do stavu k, pak význam prvku c_{ik} matice C je počet cest ze stavu i do stavu k (přes stav j). Tento význam součinu matic je jeden z důvodů, proč je násobení matic právě takto zavedeno, zdánlivě zbytečně složitě. Nabízí se otázka, proč nelze násobení matic zavést po složkách, podobně jako sčítání matic. Lze, ale takto definované násobení by mělo mnohem méně použití.

Takto zavedené matice nám umožní snadno zjistit, kolik kroků je potřeba, abychom se z daného výchozího stavu, např. ze stavu $(0;0)$ dostali do určitého stavu, např. do stavu $(5;1)$. Násobíme postupně matice P: $P^2 = P \cdot P,\ P^3 = P^2\ P,\ P^4 = P^3 \cdot P,\ \ldots$ a dostáváme

$$P^3 = \begin{pmatrix}
1&0&0&6&0&0&1&0&0&1&0&0&6&0&0&1\\
7&0&0&9&3&0&0&3&2&1&1&0&5&3&0&6\\
7&0&0&9&0&0&3&3&1&1&1&0&6&0&3&6\\
6&0&0&2&0&0&0&3&4&0&0&0&1&1&0&7\\
7&3&0&5&0&3&1&1&3&1&0&1&9&0&0&6\\
6&0&0&9&3&0&0&3&1&1&3&0&5&1&1&7\\
7&0&3&5&0&0&1&4&3&0&0&1&8&0&0&7\\
5&0&0&9&0&0&3&2&1&1&0&0&8&0&1&5\\
5&1&0&8&0&0&1&1&2&3&0&0&9&0&0&5\\
7&0&0&8&1&0&0&3&4&1&0&0&5&3&0&7\\
7&1&1&5&0&3&1&1&3&0&0&3&9&0&0&6\\
6&0&0&9&1&0&1&3&1&1&3&0&5&0&3&7\\
7&0&1&1&0&0&0&4&3&0&0&0&2&0&0&6\\
6&3&0&6&0&1&1&1&3&3&0&0&9&0&0&7\\
6&0&3&5&0&1&1&2&3&0&0&3&9&0&0&7\\
1&0&0&6&0&0&1&0&0&1&0&0&6&0&0&1
\end{pmatrix},$$

až teprve v matici

$$P^4 = \begin{pmatrix}
13 & 0 & 1 & 3 & 0 & 0 & 0 & 7 & 7 & 0 & 0 & 0 & 3 & 1 & 0 & 13 \\
20 & 6 & 0 & 19 & 0 & 4 & 3 & 5 & 10 & 5 & 0 & 1 & 25 & 1 & 0 & 21 \\
20 & 0 & 6 & 18 & 0 & 1 & 3 & 9 & 10 & 1 & 0 & 4 & 24 & 1 & 0 & 22 \\
7 & 1 & 0 & 20 & 0 & 0 & 3 & 1 & 2 & 5 & 0 & 0 & 21 & 0 & 0 & 7 \\
21 & 0 & 1 & 25 & 6 & 0 & 1 & 10 & 6 & 3 & 4 & 0 & 18 & 4 & 1 & 20 \\
21 & 4 & 1 & 18 & 0 & 6 & 3 & 5 & 10 & 2 & 0 & 4 & 25 & 1 & 0 & 20 \\
20 & 0 & 1 & 25 & 0 & 0 & 7 & 9 & 5 & 3 & 1 & 0 & 22 & 0 & 4 & 18 \\
21 & 0 & 4 & 14 & 0 & 0 & 2 & 11 & 10 & 1 & 0 & 1 & 17 & 1 & 0 & 21 \\
21 & 0 & 1 & 17 & 1 & 0 & 1 & 10 & 11 & 2 & 0 & 0 & 14 & 4 & 0 & 21 \\
18 & 4 & 0 & 22 & 0 & 1 & 3 & 5 & 9 & 7 & 0 & 0 & 25 & 1 & 0 & 20 \\
20 & 0 & 1 & 25 & 4 & 0 & 2 & 10 & 5 & 3 & 6 & 0 & 18 & 1 & 4 & 21 \\
20 & 1 & 4 & 18 & 0 & 4 & 3 & 6 & 10 & 1 & 0 & 6 & 25 & 1 & 0 & 21 \\
7 & 0 & 0 & 21 & 0 & 0 & 5 & 2 & 1 & 3 & 0 & 0 & 20 & 0 & 1 & 7 \\
22 & 0 & 1 & 24 & 4 & 0 & 1 & 10 & 9 & 3 & 1 & 0 & 18 & 6 & 0 & 20 \\
21 & 0 & 1 & 25 & 1 & 0 & 5 & 10 & 5 & 3 & 4 & 0 & 19 & 0 & 6 & 20 \\
13 & 0 & 1 & 3 & 0 & 0 & 0 & 7 & 7 & 0 & 0 & 0 & 3 & 1 & 0 & 13
\end{pmatrix}$$

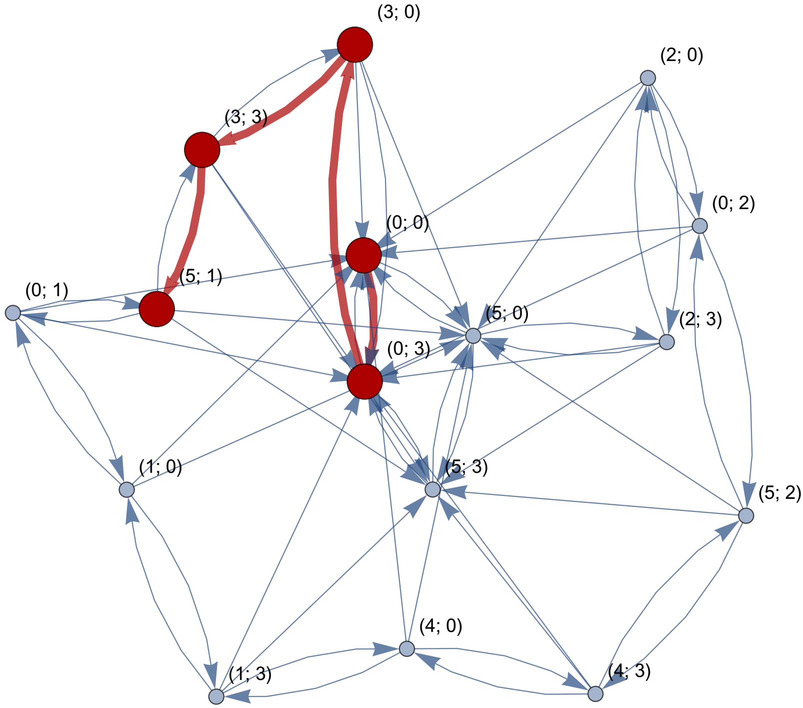
se na prvním řádku v čtrnáctém sloupci objeví poprvé nenulové číslo, které ukazuje, že se lze čtyřmi kroky dostat ze stavu $(0;0)$ do stavu $(5;1)$, který má podle původního číslování index 21 a po vyškrtnutí nedosažitelných stavů je to čtrnáctý stav.

V další kapitole si ukážeme jiný zajímavý nástroj pro sledování přechodu mezi stavy.

Teorie grafů

Body představující stavy našeho systému se v teorii grafů nazývají vrcholy (anglicky vertex). Dva vrcholy mohou být spojeny hranou (anglicky edge). Tato hrana může být orientovaná od jednoho vrcholu k druhému, jak tomu je v našem případě, kdy hrana značí možnost přechodu z jednoho stavu do druhého. Grafem pak nazýváme dvojici (V, E), kde V je množina vrcholů a E je množina hran. Hranou v orientovaném grafu rozumíme uspořádanou dvojici vrcholů grafu, viz [1, 2].

Z matice přechodů P lze sestrojit graf, viz obr. 2. V našem případě je matice přechodů P matice o velikosti 16×16, protože náš systém má 16 dosažitelných stavů. Proto bude mít graf 16 vrcholů. Každému nenulovému prvku p_{ij} matice P (těch je 58) odpovídá orientovaná hrana od i-tého vrcholu k j-tému vrcholu. Proto bude mít graf 58 orientovaných hran. Na obr. 2 jsou vrcholy označeny pro přehlednost ne indexem, ale souřadnicemi (x, y), jejichž význam je objem vody v levé a pravé nádobě.



Obr. 2: Graf popisující stavy systému dvou nádob a možné přechody mezi stavy. Tučnou čarou je vykreslena nejkratší cesta ze stavu $(0;0)$ do stavu $(5;1)$

Tučnou čarou je vykreslena nejkratší cesta ze stavu $(0;0)$ do stavu $(5;1)$. K nalezení nejkratší cesty můžeme použít Dijkstrův algoritmus, viz [3, 4]. Ten je srozumitelně popsán v pěkných skriptech [1] a tento popis algoritmu zde pro pohodlí čtenáře zopakujeme. Pro naši úlohu chceme najít nejkratší cestu z daného vrcholu do jiného daného vrcholu grafu, přičemž počítáme pouze počet operací, tedy počet hran. Dijkstrův algoritmus je obecnější ve dvou ohledech. Dovoluje nalézt nejkratší cesty z daného vrcholu do všech vrcholů grafu. Pracuje s grafy, kde hrany mohou mít různé délky a algoritmus pak hledá nejkratší cestu. To si lze představit, jako kdyby vrcholy byla města spojená silnicemi o různých délkách a my hledáme nejkratší cestu z daného města do jednotlivých měst. Tato nalezená cesta bude popsána tak, že každý vrchol bude mít přiřazeného svého předchůdce v této nejkratší cestě.

Označme v_0 výchozí vrchol, z něhož hledáme nejkratší cesty. Každému vrcholu v přiřadíme dočasnou hodnotu $dh(v)$, na počátku nekonečno, a trvalou hodnotu $th(v)$, to bude nalezená nejkratší vzdálenost od výchozího vrcholu v_0. Celý algoritmus lze popsat v těchto pěti krocích:

1. Polož $th(v_0) = 0$ a $dh(v) = \infty$ pro každý jiný vrchol.
2. Nechť u je poslední vrchol, kterému byla přiřazena trvalá hodnota (na začátku je to vrchol v_0). Pro každý vrchol v, který nemá přiřazenu trvalou hodnotu a je spojen hranou s vrcholem u, polož
$$dh(v) = \min\{dh(v), th(u) + c(u, v)\},$$
kde $c(u, v)$ je délka nebo cena hrany mezi vrcholy u a v. Pro náš případ s nádobami je cena každé existující hrany rovna jedné.
3. Pokud v předcházejícím kroku došlo ke změně dočasné hodnoty, nastav předchůdce vrcholu v vrchol u.
4. Ze všech vrcholů, které nemají přiřazenou trvalou hodnotu, vyber vrchol v, jehož dočasná hodnota $dh(v)$ je minimální, a polož $th(v) = dh(v)$. Zde je hlavní myšlenka algoritmu: Jestliže žádný jiný vrchol bez trvalé hodnoty nemá kratší vzdálenost od počátečního vrcholu, tak přes něj nemůže vést kratší cesta. To platí za důležité podmínky, že váhy hran jsou nezáporná čísla. V našem případě jsou váhy hran rovny jedné, tedy nezáporné.
5. Pokud nemají všechny vrcholy přiřazenu trvalou hodnotu, pokračuj bodem 2. Jinak algoritmus končí.

Poznamenejme, že vrcholy, které nejsou spojené s výchozím vrcholem, budou mít po dokončení Dijkstrova algoritmu délku cesty nekonečno.

Ukažme si použití Dijkstrova algoritmu na náš příklad, kdy hledáme nejkratší cestu z vrcholu $(0; 0)$ do vrcholu $(5; 1)$, viz tabulka 1. V prvním sloupci je seznam všech vrcholů ve formátu $(x; y)$. Další sloupce odpovídají stavu na začátku provádění bodu 2. algoritmu. Každé políčko obsahuje dočasnou hodnotu (nepodtržené číslo) nebo trvalou hodnotu (podtržené číslo) a za čárkou je předchůdce vrcholu, který je uveden na příslušném řádku v prvním sloupci.

Na začátku nastavíme trvalou hodnotu vrcholu $(0; 0)$ na hodnotu 0 a toto číslo podtrhneme, protože je to již trvalá hodnota. Tento bod nemá předchůdce. Ostatním vrcholů přiřadíme dočasnou hodnotu nekonečno.

Tabulka 1: Ukázka použití Dijkstrova algoritmu pro nalezení nejkratší cesty z vrcholu $(0;0)$ do vrcholu $(5;1)$. V prvním sloupci je seznam všech vrcholů. Ty odpovídají stavům systému dvou nádob. V prvním řádku jsou čísla udávající iteraci (kolo) algoritmu. V hlavní části tabulky je nepodtržené číslo dočasná hodnota, podtržené číslo je trvalá hodnota délky cesty z vrcholu $(0;0)$ do daného vrcholu (uvedeného v prvním sloupci). Za čárkou je uveden předchůdce daného vrcholu.

	1	2	3	4	5	6	7	8	9
$(0,0)$	$\underline{0}$								
$(0,1)$	∞	∞	∞	∞	∞	∞	∞	∞	∞
$(0,2)$	∞	∞	∞	∞	∞	∞	∞	$4,(2;0)$	$4,(2;0)$
$(0,3)$	∞	$\underline{1},(0;0)$							
$(1,0)$	∞	∞	∞	∞	∞	∞	∞	∞	∞
$(1,3)$	∞	∞	∞	∞	∞	∞	∞	∞	∞
$(2,0)$	∞	∞	∞	∞	$3,(2;3)$	$3,(2;3)$	$\underline{3},(2;3)$		
$(2,3)$	∞	∞	∞	$\underline{2},(5;0)$					
$(3,0)$	∞	∞	$2,(0;3)$	$2,(0;3)$	$\underline{2},(0;3)$				
$(3,3)$	∞	∞	∞	∞	∞	$3,(3;0)$	$3,(3;0)$	$\underline{3},(3;0)$	
$(4,0)$	∞	∞	∞	∞	∞	∞	∞	∞	∞
$(4,4)$	∞	∞	∞	∞	∞	∞	∞	∞	∞
$(5,0)$	∞	$1,(0;0)$	$\underline{1},(0;0)$						
$(5,1)$	∞	∞	∞	∞	∞	∞	∞	∞	$\underline{4},(3;3)$
$(5,2)$	∞	∞	∞	∞	∞	∞	∞	∞	∞
$(5,3)$	∞	∞	$2,(0;3)$	$2,(0;3)$	$2,(0;3)$	$\underline{2},(0;3)$			

To zapíšeme do prvního sloupce v hlavní části tabulky, který má nahoře číslo 1.

V 1. kole algoritmu je $u=(0;0)$. Ten je spojen se dvěma vrcholy, $(0;3)$ a $(5;0)$. Proto nastavíme jejich dočasnou hodnotu 1 a jejich předchůdce $(0;0)$. Ve 4. bodě vybereme např. vrchol $(0;3)$ a jeho dočasnou hodnotu prohlásíme za trvalou (v tabulce ji podtrhneme). Existuje-li více vrcholů s minimální dočasnou hodnotou, vybereme libovolný z nich. Na ostatní se pak dostane při dalších kolech algoritmu.

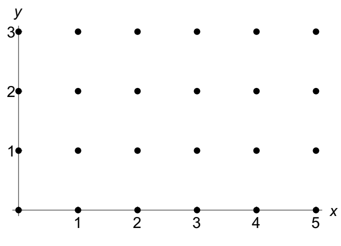
V 2. kole algoritmu je $u=(0;3)$. Ten je spojen se třemi vrcholy, $(5;3)$, $(3;0)$ a $(0;0)$. Vrchol $(0;0)$ má již nastavenu trvalou hodnotu, proto jeho záznam již neměníme (ani neopisujeme do dalších sloupečků). Pro vrcholy $(5;3)$ a $(3;0)$ nastavíme jejich dočasnou hodnotu 2 a jejich předchůdce $(0;3)$. Záznam vrcholu $(5;0)$ opíšeme. Minimum dočasných hodnot má nyní vrchol $(5;0)$, a to 1. Proto tuto jedničku prohlásíme za trvalou hodnotu a v tabulce si ji podtrhneme.

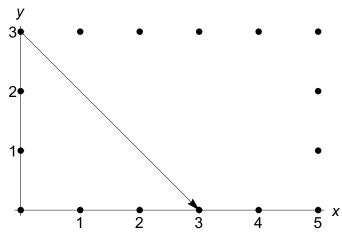
Ve 3. kole algoritmu je $u = (5;0)$. Ten je spojen se třemi vrcholy, $(0;0)$, $(5;3)$ a $(2;3)$. Vrchol $(0;0)$ již má přiřazenu trvalou hodnotu, a to 0, proto s ním již nepracujeme. Do vrcholu $(5;3)$ vedou dvě cesty: původní přes vrchol $(0;3)$ a nová přes vrchol $(5;0)$, obě stejně dlouhé (2), proto jeho údaje nyní neměníme. Do vrcholu $(2;3)$ jsme zatím našli cestu délky 2, jeho předchůdce je zatím vrchol $(5;0)$. Z třech vrcholů $(2;3)$, $(3;0)$ a $(5;3)$ s dočasnou hodnotou 2 vybereme např. vrchol $(2;3)$ a prohlásíme jeho dočasnou hodnotu za trvalou.

Takto postupuje dále, výsledky zapisujeme do tabulky, až v 9. kole dospějeme k výsledku, že do bodu $(5;1)$ vede cesta délky 4 a přečtením předchůdců jednotlivých bodů zjistíme, jak tato nejkratší cesta vypadá, tedy přes které vrcholy vede.

Příkaz pro hledání nejkratší cesty je běžnou součástí softwarových nástrojů pro práci s grafy. Např. v prostředí *Mathematica* se jmenuje `FindShortestPath`.

Modulární aritmetika

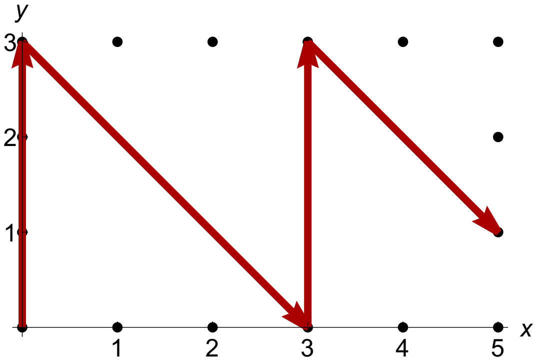




Obr. 3: Vlevo: body v rovině představující všechny možné stavy soustavy dvou nádob. Vpravo: pouze dosažitelné stavy. Šipkou je zobrazen přechod z bodu $(0;3)$ do bodu $(3;0)$, tedy kdy všechnu vodu z plné malé nádoby přelijeme do prázdné velké nádoby

Podívejme se na ještě jeden možný způsob řešení naší úlohy o přelévání vody mezi dvěma nádobami. Když si množství vody v levé nádobě označíme x a množství vody v pravé nádobě y, můžeme stav našeho systému dvou nádob vyjádřit bodem v rovině o kartézských souřadnicích $(x;y)$.

Všechny možné stavy pak vytvoří obdélníkovou síť bodů o celočíselných souřadnicích (viz obr. 3 vlevo) a dosažitelné stavy jsou reprezentovány body na obvodu tohoto obdélníku (viz obr. 3 vpravo).



Obr. 4: Nejkratší cesta z bodu $(0;0)$ do bodu $(5;1)$ podobně jako na obr. 2, ale zde jsou vrcholy grafu umístěny do bodů podle jejich kartézských souřadnic

Jaké přechody jsou možné a jak se zobrazí na obrázku? Vyjděme ze stavu, kdy jsou obě nádoby prázdné, tedy z bodu $(0;0)$. Můžeme naplnit velkou nádobu vodou. Tím se přesuneme z levého dolního rohu do pravého dolního rohu. Nebo můžeme naplnit malou nádobu, pak se přesuneme do levého horního rohu. Když naplníme i druhou nádobu, tak se přesuneme do pravého horního rohu. Vyprázdnění velké nádoby znamená přesun na levý okraj (malá nádoba zůstává beze změny, tedy souřadnice y se nemění). Vyprázdnění malé nádoby znamená přesun na dolní okraj (nyní velká nádoba zůstává beze změny, tedy souřadnice x se nemění). Podobně naplnění velké nádoby je přesun na pravý okraj, stav malé nádoby se nemění, tedy souřadnice y se nemění. Naplnění malé nádoby je přesun na horní okraj při zachování souřadnice x.

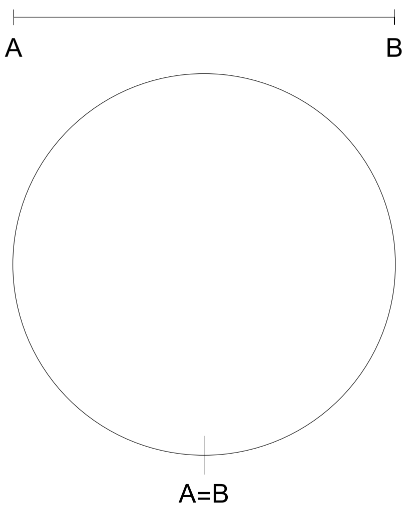
A co přelévání z jedné nádoby do druhé? Při tomto ději se celkové množství vody v obou nádobách zachovává, pohybujeme se po přímce $x + y =$ konst, to je přímka klesající pod úhlem 45°. Na obr. 3 vpravo je šipkou mířící doprava dolů zobrazen přechod, kdy přelijeme veškerou vodu z plné malé nádoby do velké nádoby.

Obr. 4 je obdobou obr. 2, tedy ukazuje dosažitelné body a nejkratší cestu z bodu $(0;0)$ do bodu $(5;1)$, ale vrcholy jsou umístěny v rovině podle jejich kartézských souřadnic.

Jak je to s dosažitelností jednotlivých bodů? Při přelévání vody z malé nádoby do velké (pohyb na obr. 4 doprava dolů) proces končí, když buďto dojde voda v malé nádobě (na obr. 4 narazíme na dolní okraj) nebo dosáhneme plné velké nádoby (na obr. 4 narazíme na pravý okraj). A podobně při přelévání vody z velké nádoby do malé (pohyb na obr. 4 doleva nahoru) proces končí, když buďto dojde voda ve velké nádobě (na obr. 4 narazíme na levý okraj) nebo dosáhneme plné malé nádoby (na obr. 4 narazíme na horní okraj). Při hledání všech dosažitelných bodů můžeme tedy sledovat, kam se všude dostaneme, když provádíme přelévání z malé nádoby do velké (pohyb doprava dolů) doprovázené doléváním malé nádoby, když je prázdná, a vyléváním velké nádoby do odpadu, když je plná. Nebo obráceně přeléváme vodu z velké nádoby do malé, doplnění doléváním velké nádoby a vyprazdňováním malé.

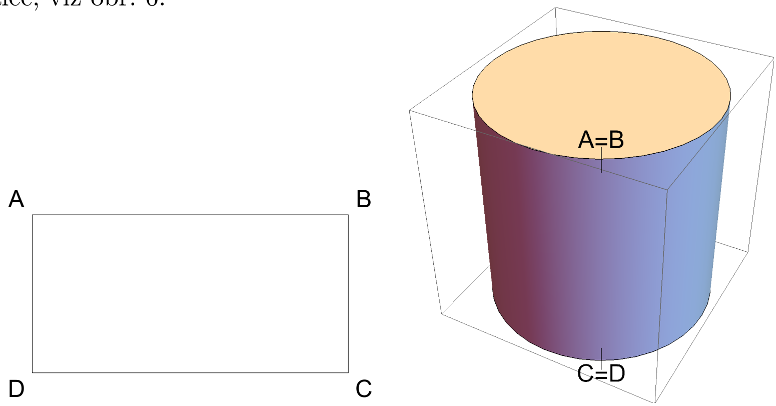
Ty pomocné operace (doplňování a vylévání vody) vlastně jen ztotožňují body na horním okraji s body na dolním okraji obdélníku a body na levém okraji s body na pravém okraji.

Když ztotožníme krajní body úsečky, dostaneme kružnici (až na nepodstatné deformace), viz obr. 5.



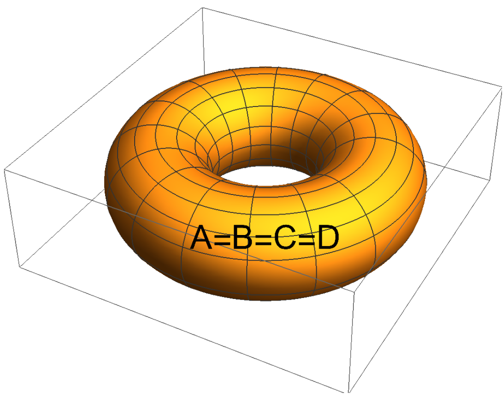
Obr. 5: Když ztotožníme krajní body úsečky (jako bychom zapnuli pásek u kalhot), dostaneme kružnici

Když ztotožníme hranu obdélníku s protilehlou hranou, dostaneme válec, viz obr. 6.



Obr. 6: Když ztotožníme levou stranu obdélníku s protější stranou (jako bychom stočili vysokoškolský diplom do tuby), dostaneme válec

A když ztotožníme navíc druhou stranu obdélníku s protilehlou stranou, dostaneme torus (ten si můžeme představit jako povrch nafouknuté pneumatiky), viz obr. 7. Milovníci sladkého pečiva si místo pneumatiky raději představí americkou koblihu neboli donut.



Obr. 7: Torus je plocha, kterou si můžeme představit jako dvakrát stočený obdélník

Takže přelévání vody z malé nádoby do velké, doprovázené doplňováním vody do malé nádoby, když je prázdná, a vyléváním vody z velké nádoby, když je plná, si můžeme představit jako pohyb jihovýchodním směrem v rovině. A buďto uvažujeme, že opustíme náš základní obdélník a dostáváme se postupně do jeho dalších kopií, nebo se při nárazu na pravou hranu vrátíme na levou hranu a při nárazu na spodní hranu se vrátíme na horní hranu. To je ekvivalentní nahrazení vodorovné souřadnice x výrazem $x \bmod 5$ a nahrazení svislé souřadnice y výrazem $y \bmod 3$, kde mod je zbytek po dělení. Např. $11 \bmod 5 = 1$, protože $11 : 5$ je 2 a zbude 1. Anebo se na tento proces můžeme dívat jako na pohyb po toru.

Naše otázka zní, které všechny body lze takto navštívit. Průsečíky přímky popisující sledovaný proces s dolní hranou obdélníku, přesněji s body o vodorovné souřadnici $x \in \{0, 1, 2, 3, 4\}$ se střídají tak, že nový průsečík je o 3 jednotky vpravo oproti předchozímu průsečíku. Navštívíme takto všechny body?

Je-li počet bodů, zde 5, nesoudělný s velikostí kroku, zde 3, pak navštívíme všechny body. To lze snadno dokázat sporem. Uvažujme obecně počet bodů a (zde 5) na úsečce a velikost skoku b (zde 3). Předpokládáme, že přirozená čísla a a b jsou nesoudělná, tedy jejich největší společný dělitel $\gcd(a, b) = 1$. Zkratka gcd pochází z anglického výrazu greatest common divisor. Dále předpokládáme, že když začneme z bodu x_0, tak dostáváme postupně body

$$x_1 \equiv x_0 + b \mod a, \qquad x_2 \equiv x_0 + 2b \mod a,$$

obecně

$$x_n \equiv x_0 + nb \mod a.$$

Chceme dokázat, že takto navštívíme všechny body množiny $\{0, 1, 2, \ldots, a-1\}$. Důkaz provedeme sporem. Předpokládejme tedy, že po n krocích se dostaneme do bodu, kde jsme již byli po m krocích, aniž bychom mezitím navštívili všechny body. Pak by se navštívené body opakovaly a dosud nenavštívené body by nebyly navštíveny nikdy. Přitom má platit $0 < n - m < a$. Dostat se do bodu, kde jsme již byli, znamená

$$x_m \bmod a = x_n \bmod a,$$

tedy

$$x_n - x_m = ka,$$

kde $k \in N$ je přirozené číslo. Protože

$$x_m \equiv x_0 + mb \mod a, \qquad x_n \equiv x_0 + nb \mod a,$$

je

$$x_n - x_m = (n - m)b = ka.$$

Zde jsou čísla a a b nesoudělná, takže rozdíl $(n-m)$ musí být násobkem a. To ale není možné, protože $0 < n - m < a$. Tím je důkaz proveden.

To znamená, že jsou-li objemy nádob a a b nesoudělná čísla, jako tomu je v našem původním příkladu, kde $a = 5$ a $b = 3$, tak postupně navštívíme všechny body obvodu obdélníku.

Kdyby naopak čísla a a b nebyla nesoudělná, například kdyby bylo $a = 10$ a $b = 6$, tedy jejich největší společný dělitel $\gcd(a, b) = 2$, tak by dosažitelné body byly pouze ty body, jejichž souřadnice jsou násobky čísla 2, nikdy bychom nedostali bod, kde $x = 1$.

A kdyby dokonce podíl a/b nebyl racionální, ale iracionální číslo, tak by pohyb po toru vytvořil křivku, která by jeho povrch vyplnila hustě a nikdy se nevrátila do bodu, ve kterém již byla. Výrazem hustě vyplnit zde myslíme to, že v libovolně malém okolí libovolného bodu se nachází bod křivky.

Závěr

Na jednoduché úloze jsme ukázali, jak zajímavé matematické nástroje můžeme použít. Chvílemi se mohlo zdát, že jdeme louskat oříšky parním strojem, ale pro nás to byla dobrá příležitost seznámit se s těmito užitečnými postupy.

Literatura

[1] Turzík, D., Pavlíková, P.: *Diskrétní matematika.* VŠCHT, Praha, 2007.

[2] Wikipedia: *Graph (discrete mathematics).* [online] `https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)`.

[3] Wikipedia: *Dijkstra's algorithm.* [online] `https://en.wikipedia.org/wiki/Dijkstra's_algorithm`.

[4] Kára, J., Král', D., Mareš, M.: Recepty z programátorské kuchařky Korespondenčního semináře z programování, 1. část. *Rozhledy matematicko-fyzikální*, 80 (2005), č. 1, s. 26–33.