Cvičení Programování I

Cvičící: **Pavel Surynek, KTIML**
`surynek@ktiml.mff.cuni.cz`
`http://ktiml.mff.cuni.cz/~surynek`
Semestr: **Zima 2007/2008**
Kroužek: **Matematika/53**
Rozvrh: **Středa 12:20-13:50 (učebna K7)**

Stručné poznámky ke cvičení z 7.11.2007

0. Organizační záležitosti. Za domácí úkol byly úlohy **jak dlouho může běžet program**, **pigeon-hole principle**, dále byly programovací úlohy (to, co už jsme vyřešili slovně) - **sedlový bod**, **největší společný dělitel** a **hledání prvku v setříděné posloupnosti**.

1. NSD pascalovsky. Naprogramujte v Pascalu program na výpočet největšího společného dělitele dvou čísel.

2. Sedlový bod pascalovsky. Je dána čtvercová matice nad celými čísly. *Napište program v Pascalu*, který co nejrychleji v této matici nalézt sedlový bod. Sedlový bod je místo v matici, které obsahuje největší hodnotu ve svém řádku a nejmenší hodnotu ve svém sloupci nebo nejmenší hodnotu ve svém řádku a největší hodnotu ve svém sloupci. Snažte se minimalizovat počet kroků, přičemž za krok považujeme každé podívání se na políčko matice. Nápověda: předvýpočet.

3. Vyhledávání s setříděné posloupnosti pascalovsky. Je dána setříděná posloupnost N přirozených čísel (například: 5, 20, 27, 35, 60, 66, 91). *Napište program v Pascalu*, který pro dané přirozené číslo x (například: 26) a zmiňovanou posloupnost a rozhodne, zda se zadané přirozené číslo nachází v zadané posloupnosti (v uvedeném příkladě bude odpověď ne). Algoritmus může provádět operace porovnání a v každém kroku se „může podívat“ na jedno číslo na libovolné pozici v posloupnosti. Snažte se minimalizovat počet kroků algoritmu (porovnání, podívání se, ... se považuje za krok).

4. Souvislost grafu maticově. Vsuvka: Násobení matic. Nechť A je matice typu $m \times n$ nad přirozenými čísly

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \text{ kde } a_{ij} \in N \text{ pro } i \in \{1,2,...,m\},\ j \in \{1,2,...,n\}$$

nechť B je matice typu $n \times k$ nad přirozenými čísly

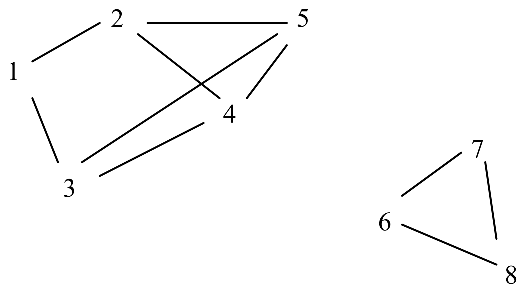
$$B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1k} \\ b_{21} & b_{22} & \dots & b_{2k} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nk} \end{pmatrix}, \text{ kde } b_{ij} \in N \text{ pro } i \in \{1,2,...,n\},\ j \in \{1,2,...,k\}.$$

Součinem matic A a B je matice C typu $m \times k$ nad přirozenými čísly

$$A \times B = C = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1k} \\ c_{21} & c_{22} & \dots & c_{2k} \\ \vdots & \vdots & & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mk} \end{pmatrix}, \text{ kde } c_{ij} = \sum_{l=1}^{n} a_{il} b_{lj} \text{ pro } i \in \{1,2,\dots,m\},\ j \in \{1,2,\dots,k\}.$$

Násobení matic lze snadno definovat též pro Booleovské matice, tj. matice nad $\{0,1\}$, resp. $\{pravda, nepravda\}$. Stačí nahradit operaci součtu operací logického součtu a operaci součinu operací logického součinu.

Je dán neorientovaný graf $G = (V, E)$, kde $V = \{1,2,\dots,n\}$, pro n přirozené číslo, a $E \subseteq \binom{V}{2}$. V je množina vrcholů a E je množina hran. Graf lze znázornit diagramem, jako například na následujícím obrázku (vrcholy jsou vrcholy a hrany spojnice mezi vrcholy):



Uvedený graf lze jednoznačně reprezentovat pomocí Boolovské matice M typu $n \times n$, kde $m_{ij} = 1$, právě když $\{i, j\} \in E$ a $m_{ij} = 0$ jinak, a toto platí pro každé $i, j \in \{1,2,\dots,n\}$. Matice M se nazývá *matice sousednosti* grafu G. V matici $M' = M \times M$ je $m'_{ij} = 1$, právě když mezi vrcholy i a j vede cesta délky 2, $m'_{ij} = 0$ jinak, a toto platí pro každé $i, j \in \{1,2,\dots,n\}$.

Navrhněte co nejefektivnější algoritmus a zformulujte jej jako program v Pascalu, který odpoví na dotaz (nebo více dotazů), zda mezi vrcholy i a j vede cesta. Využijte přitom násobení matic a snažte se přitom počet těchto násobení minimalizovat.

5. Hromada. Hromada, přesněji 2-regulární hromada, je binární strom splňující následující podmínky:

i. (trochu neformálně) Strom je zaplňován zleva, tj. vždy vypadá následovně:

ii. Uzly stromu jsou ohodnoceny a platí, že ohodnocení synů vrcholu v je větší (větší nebo rovno) než ohodnocení vrcholu v (jejich otce).

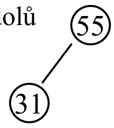
Vsuvka: Co je to strom? Strom je souvislý graf bez kružnic. Co je to graf? Neformálně: graf jsou vrcholy, z nichž některé jsou po dvojicích pospojované spojnicemi (čárami). Formálněji graf G je dvojice (V, E), kde V je množina vrcholů a $E \subseteq \binom{V}{2}$ je množina hran. Přičemž $\binom{V}{2}$ značí všechny dvojice vrcholů. Souvislost znamená, že je možné dostat se z libovolného vrcholu přes hrany do libovolného jiného vrcholu. Graf obsahuje kružnici, když mezi dvěma vrcholy existují dvě různé cesty, jak se z jednoho dostat skrz hrany do druhého.

Definici a konstrukci haldy objasníme na příkladu. Budeme chtít do haldy vložit prvky 55, 31, 4, 27, 99, 42 a 2.

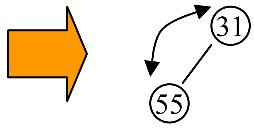
1. Vložíme prvek 55 „dolů a doleva“

Všechny podmínky jsou splněny

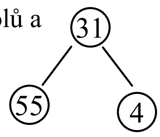
2. Vložíme prvek 31 „dolů a doleva“



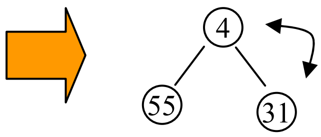
Podmínka na tvar stromu je splněna, ale není splněna podmínka na uspořádání



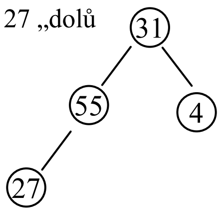
3. Vložíme prvek 4 „dolů a doleva“



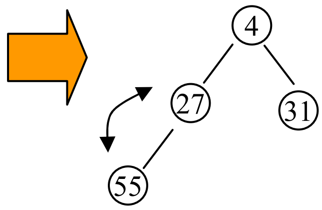
Podmínka na tvar stromu je splněna, ale není splněna podmínka na uspořádání



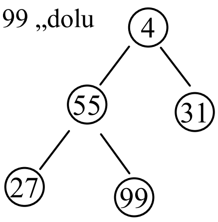
4. Vložíme prvek 27 „dolů doleva“



Podmínka na tvar stromu je splněna, ale není splněna podmínka na uspořádání

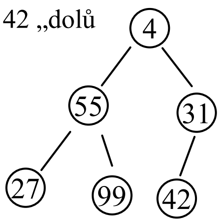


5. Vložíme prvek 99 „dolu a doleva“



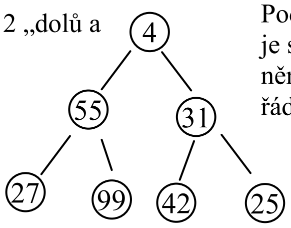
Všechny podmínky na haldu jsou splněny

6. Vložíme prvek 42 „dolů doleva“

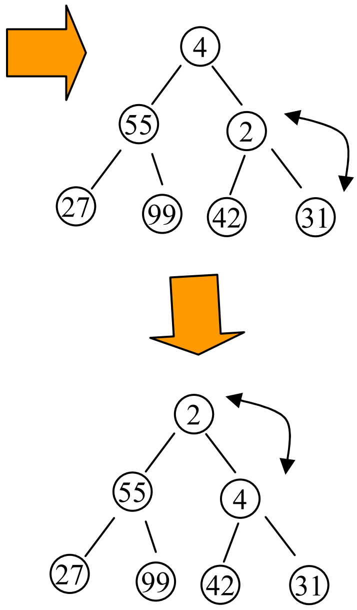


Všechny podmínky na haldu jsou splněny

7. Vložíme prvek 2 „dolů a doleva“

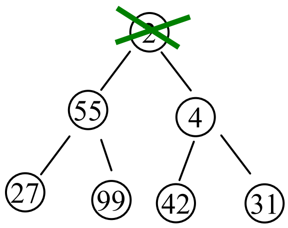


Podmínka na tvar stromu je splněna, ale není splněna podmínka na uspořádání



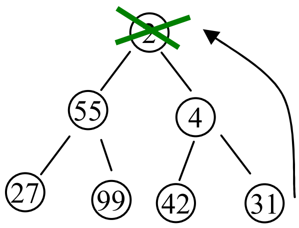
Při konstrukci haldy postupujeme tak, že vytvoříme uzel stromu do ještě nezaplněné hladiny co nejvíce doleva, tam vložíme nový prvek. Tím se ale může porušit podmínka na uspořádání. Tu opravíme tak, že prohodíme otce a prvek nový prvek. Tím ale můžeme porušit podmínku na uspořádání pro otce nově vytvořeného vrcholu. Opět podmínku opravíme stejným způsobem a pokračujeme tak dlouho dokud se podmínku na uspořádání nepodaří pro předchůdce změněného vrcholu splnit nebo nedojdeme ke kořeni stromu.

Evidentně platí, že v kořeni stromu je nejmenší prvek. Pokusme se toho využít pro setřídění prvků. Odeberme kořen haldy a jeho hodnotu zařaďme do setříděné posloupnosti.

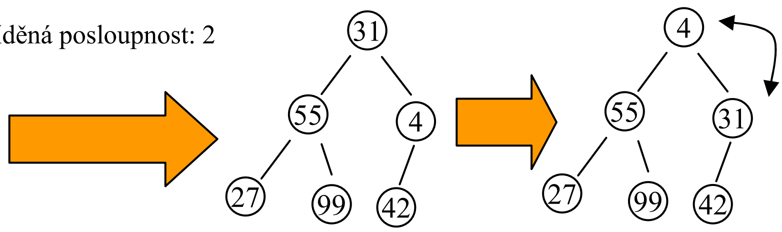


Setříděná posloupnost: 2

Nyní je potřeba haldu opravit. To se provede tak, že do kořene přesuneme nejpravější vrchol z nejspodnější hladiny. Tím zajistíme splnění podmínky na tvar haldy, ale poruší se tím podmínka na uspořádání. Podmínku na uspořádání opravíme postupným snižováním nového kořene směrem k synovi s menším ohodnocením. To provádíme tak dlouho dokud není podmínka na uspořádání splněna.



Setříděná posloupnost: 2



Vaším úkolem je spočítat počet kroků, jež je potřeba na setřídění posloupnosti N čísel. Za krok se považuje jakákoli elementární operace. Představte si, že úlohu řešíte pomocí tužky a papíru, elementární operací je pak třeba nakreslení čárky.

6. Křivka v rovině. Mějme křivku v rovině zadanou parametricky, například:

$$x(t) = a_x t^4 + b_x t^3 + c_x t^2 + d_x t + e_x$$

$$y(t) = a_y t^4 + b_y t^3 + c_y t^2 + d_y t + e_y,$$ kde $t \in [0,1]$ a a_x, b_x, c_x, d_x, e_x, a_y, b_y, c_y, d_y a e_y jsou reálné konstanty. Úkolem je křivku „bod po bodu“ nakreslit, tj. zvolíme nějaký krok pro parametr t, například 0.001, a postupně počítáme $x(t)$ a $y(t)$ pro $t = 0.0$, $t = 0.001$, $t = 0.002$, $t = 0.003$ atd. a výsledné body vykreslujeme. Musíme tedy $1000\times$ vypočítat hodnotu výrazů pro $x(t)$ a $y(t)$. Navrhněte schéma pro výpočet $x(t)$ a $y(t)$, které minimalizuje počet aritmetických operací. Uvažte, že operace sčítání je někdy rychlejší než operace násobení.

7. Křivka v rovině pascalovsky. V nápovědě v Pascalu prostudujte práci s grafikou, jmenovitě, jak zapnout grafický režim a proceduru PutPixel, případně proceduru Line. Úlohu křivka v rovině naprogramujte.