Diskrétní matematika – cvičení

Stromy – návod

Úkol 1: Skóre stromů

Rozhodněte, zda posloupnost (1, 1, 1, 1, 2, 3, 3, 4) může být skóre stromu. *Rada*: Nejprve zjistěte, zda daná posloupnost je skóre grafu, a pak diskutujte vzájemný vztah mezi počtem vrcholů a hran.

Řešení:

Nejdřív jako předtím (předchozí cvičení, kde jsme určovali, zda je to skóre a kreslili k dané posloupnosti graf, resp. dva neizomorfní grafy) , určíme nejdřív, zda daná posloupnost je skóre grafu:

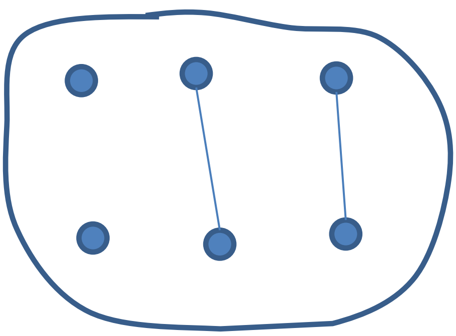
D=(1, 1, 1, 1, 2, 3, 3, 4) % poslední největší hodnotu v uspořádané posloupnosti odebereme a předchozí hodnoty (počet určuje odebraná hodnota) snížíme o 1

D^{i}=(1, 1, 1, 0, 1, 2, 2) % postup opakujeme, ale nezapomeňme, že posloupnost musíme vždy uspořádat !!!

D^{i}= (0, 1, 1, 1, 1, 2, ~~2~~) % je to jenom přeuspořádána posloupnost D´ a opět opakujeme, odstraníme poslední hodnotu a předchozí (počet odebrané hodnoty) snížíme o 1

D^{ii}=(0, 1, 1, 1, 0, 1) % přeuspořádáme

D^{ii}=(0, 0, 1, 1, 1, 1) % kdo už vidí, že daná posloupnost je skóre, protože už k ní dokáže nakreslit graf, viz níž (odpovídá skóre 0,0,1,1,1,1):



Tak může přestat tady, nakreslit daný graf z obrázku, čímž potvrdí, že D´´ je skóre a podle probírané věty z přednášky je skóre i D.

Když to někdo nevidí, může pokračovat dál a dojít k nulám, čímž potvrdí, že je to skóre, nebo k záporné hodnotě, co napovídá, že to skóre není.

Tady pro všechny studenty, kteří chtějí jít až nulám, pokračuji dál:

D^{ii}=(0, 0, 1, 1, 1, ~~1~~)

$D^{iii}=(0, 0, 1, 1, 0) \rightarrow D^{iii} = (0, 0, 0, \underline{1}, \cancel{1})$

$D^{iv}=(0, 0, 0, 0)$ …. Všechny nuly, což odpovídá grafu se čtyřmi izolovanými vrcholy, proto D^{iv} je skóre grafů a podle vzpomínané věty i D = (1, 1, 1, 1, 2, 3, 3, 4) je skóre grafu.

Máme první část. Teď druhou: potřebuje zjistit, zda to může být skóre stromu. Podle vlastností stromů, které jsme probírali na přednášce, tak víme, že pro každý graf, který je stromem musím platit následující vztah mezi počtem hran a vrcholů: m=n-1 (počet vrcholů je o jedno víc než hran/počet hran je o jedno míň než vrcholů). Zjistíme počty daného skóre:

n = 8 % jenom sečteme počet stupňů v posloupnosti, to odpovídá počtu vrcholů

m = 16/2=8 %použijeme větu o sudosti, součet všech stupňů je roven 2 krát počet hran

- Dosadíme do vztahu o počtu hran a vrcholů: m = n – 1

$8 \neq 8 - 1$ % protože rovnost je porušena, daná posloupnost nemůže být skórem stromu, může být jenom, kdyby platila rovnost

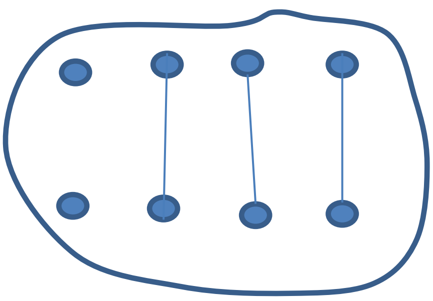
b) analogicky (1,1,1,1,1,1,1,1,1,1,2,2,3,5)

c) (1,1,1,1,1,1,2,3,3,4)
Řešení:

$D = (1,1,1,1,1,\underline{1,2,3,3},4)$

$D^{i}= (1,1,1,1,1,0,1,2,2) \rightarrow D^{i}= (0,1,1,1,1,1,\underline{1,2},\cancel{2})$

$D^{ii}=(0,1,1,1,1,1,0,1) \rightarrow D^{ii}=(0,0,1,1,1,1,1,1)$ %můžeme končit, jde o graf s dvěma izolovanými vrcholy a třemi hranami



D^{ii} je skóre grafu, teda i D = (1,1,1,1,1,1,2,3,3,4) je skóre grafu.

Teď zjistíme, zda to může být skóre stromu:

m = 18/2 = 9

n = 10

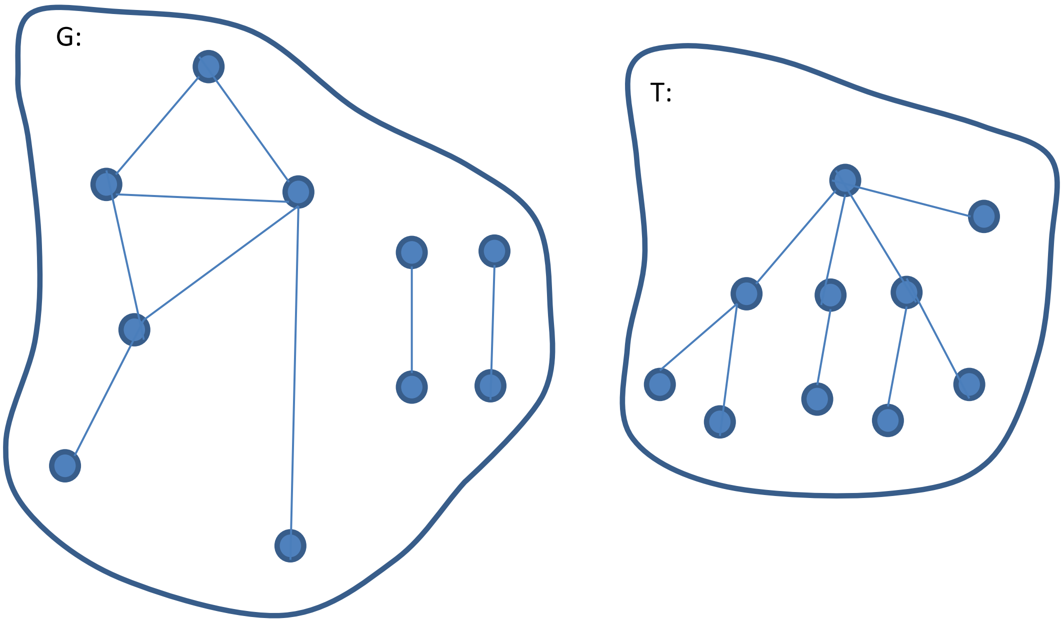
dosadíme do vztahu: m=n-1

9 = 10 – 1

9 = 9

Rovnost platí ⇒ posloupnost (1,1,1,1,1,1,2,3,3,4) může být skóre stromu.

% slovíčko může je důležité, protože může existovat i graf s daným skóre, který není strom, viz ukázka grafů níž



d) (1,1,1,1,1,2,2,3,3,3)

e) (1,1,1,1,1,1,2,2,3,5)

f) (1,1,1,1,1,1,1,1,2,3,4,5)

g) (1,1,1,1,1,1,2,2,2,2,3,4,5)

Úkol 2: Strom?

Uděláte analogicky jako úkol 1, napíšete si ze zadání seznam stupňů jako posloupnost a pak řešíte jenom druhou část (první nemusíte, protože v zadání máte dané, že G je souvislý graf, to znamená, že ta posloupnost už bude skóre souvislého grafu), tj. zjišťujete, zda sedí vztah m=n-1.

1) Nechť graf *G* je souvislý graf, který má 12 vrcholů stupně 1, tři vrcholy stupně 2 a čtyři vrcholy stupně 5. Určete a zdůvodněte, zda graf *G* je či není strom.

2) Nechť graf *G* je obyčejný souvislý graf, který má 10 vrcholů stupně jedna, čtyři vrcholy stupně dva a čtyři vrcholy stupně čtyři. Určete a zdůvodněte, zda graf *G* je nebo není strom.

Úkol 3: Skóre souvislého grafu

Opět analogicky jako úkol 1, jenom budete ověřovat vztah: $m \geq n - 1$. Strom je graf, který je souvislý graf s nejmenším počtem hran. Když odeberete jednu hranu ze stromu, tak se stane nesouvislým grafem. Proto souvislý graf musí mít minimálně n-1 hran. Není to však postačující podmínka, to znamená, že když tento vztah platí, neznamená to, že musí být graf souvislý. Ale když graf je souvislý, tak určitě má minimálně n-1 hran.

Rozhodněte, zda posloupnosti

(2, 2, 2, 3, 3, 4, 4)

(1,1,1,1,1,1, 2, 2, 2)

můžou být skóre souvislého grafu. Své rozhodnutí zdůvodněte.

Rada: Nejprve zjistěte, zda posloupnost je skóre grafu. Pak zjistěte, zda graf může být souvislý.

Úloha 3: Les

1. Kolik komponent obsahuje les s 12 vrcholy a 8 hranami?

Řešení:
Pro les platí: je to acyklický graf. Co se týče počtu hran a vrcholů platí vztah: m = n – k, kde k je počet komponent.
Máme dané: n = 12, m = 8. Dosadíme do vztahu a vypočítáme k: 8 = 12 – k $\Rightarrow$ k = 4.
Odpověď: Les má 4 komponenty.

Analogicky řešit zbytek:

2. Kolik hran má les obsahující 7 vrcholů a 3 komponenty?

3. Kolik komponent obsahuje les s 13 vrcholy a 11 hranami?

4. Kolik vrcholů má les obsahující 10 hran a 4 komponenty?

5. Kolik hran a kolik komponent má les, jehož skóre je (1, 1, 1, 1, 1, 1, 1, 1, 2, 3, 3, 4)?

Úloha 4: Negace a obměna

Toto nechám na Vás:

K následujícím tvrzením utvořte negaci a obměnu:

1) $\forall_{G=(V,E)}$ *G* je strom $\Rightarrow$ každá hrana grafu *G* je most

2) $\forall_{G=(V,E)}$ *G* je strom $\Rightarrow$ pro každé dva vrcholy *x*, *y* grafu *G* existuje právě jedna cesta mezi vrcholy *x* a *y*

Úloha 5: Vyvraťte tvrzení

Opět nechám na vás, nejdřív znegujte a pak podle negace se pokuste nakreslit graf, který bude vyvracet původní tvrzení:

1) $\forall_{G=(V,E)}$ každá hrana grafu *G* je most $\Rightarrow$ *G* je strom

2) $\forall_{G=(V,E)}$ v grafu G platí vztah $|V| = |E| + 1 \Rightarrow G$ je strom

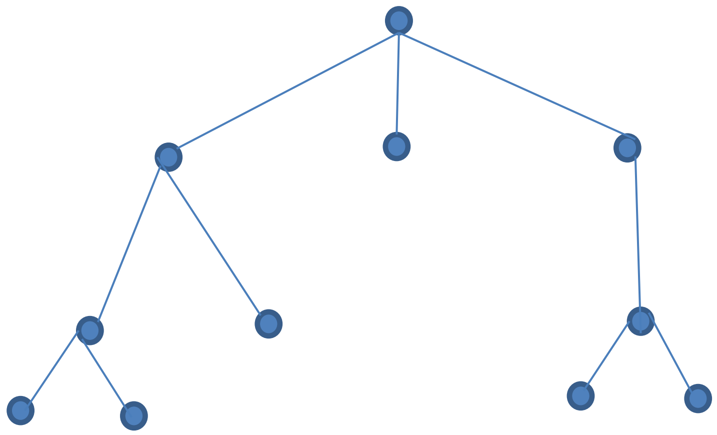
Úloha 6: Izomorfismus stromů

Opět nejdřív nechám na vás (rada: když budete kreslit neizomorfní stromy, kreslete systematicky, je jich 6):

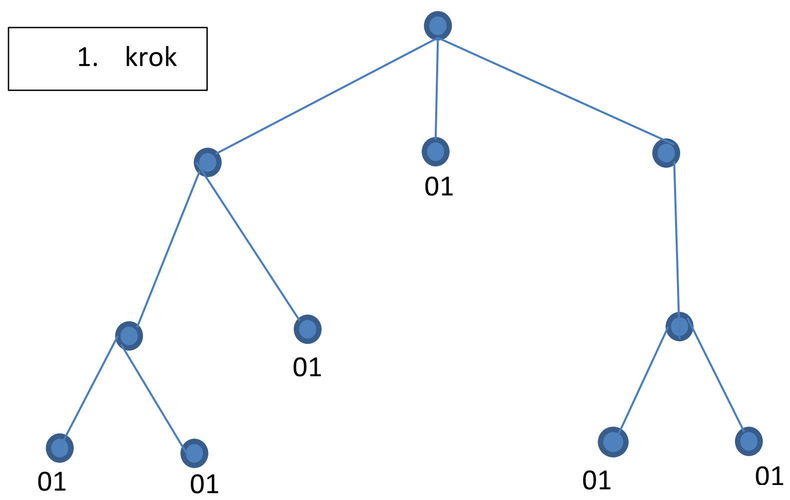
- Určete dva stromy, které mají 7 vrcholů, mají stejné skóre, a přesto nejsou izomorfní.
- Nakreslete všechny neizomorfní stromy na 6 vrcholech.

Úloha 7: Kód uspořádaného kořenového stromu

Ukázka: Napište kód následujícího uspořádaného stromu:

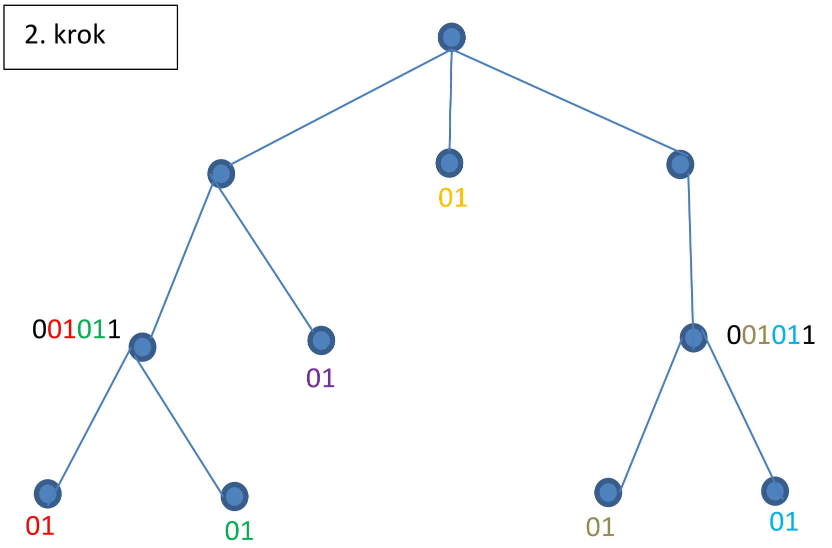


Nejdřív každý list dostane kód

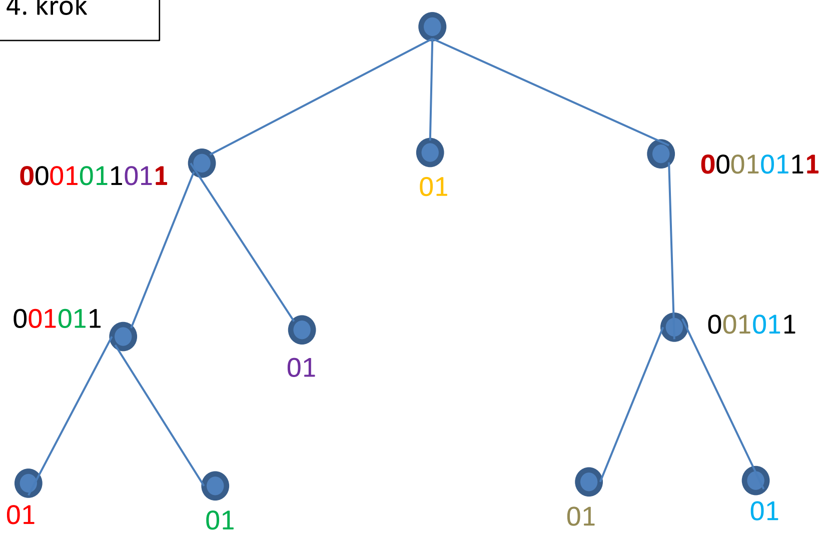


Pak budeme přiřazovat kódy ostatním vrcholům. Postupujeme od spodních vrstev, nejspodnější už je. Přímý předchůdce zdědí kódy svých přímých potomků v pořadí z leva doprava, a pak připíšeme na začátek 0 a na konec 1.

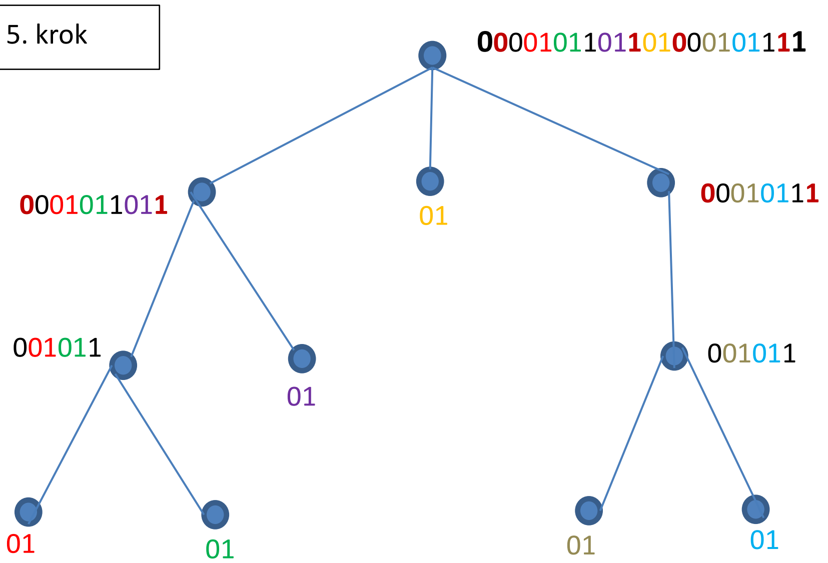
2. krok



4. krok

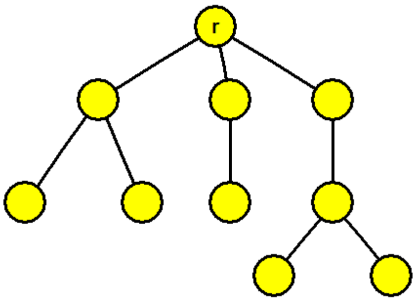


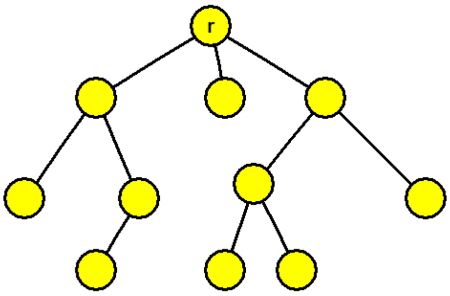
5. krok



Analogicky okódujte následující stromy:

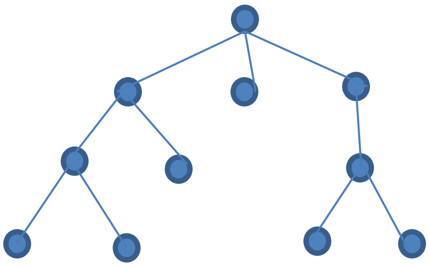
Napište kódy následujících uspořádaných kořenových stromů:



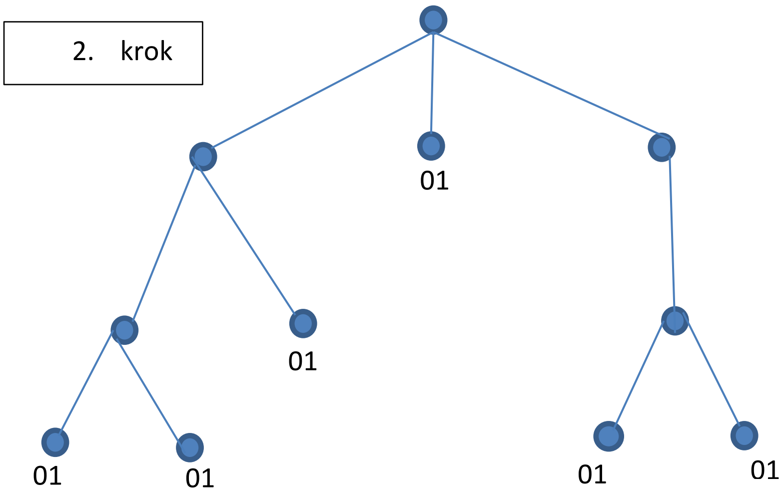


Úloha 7: Minimální kód uspořádaného kořenového stromu

Ukázka: Napište minimální kód následujícího uspořádaného stromu:

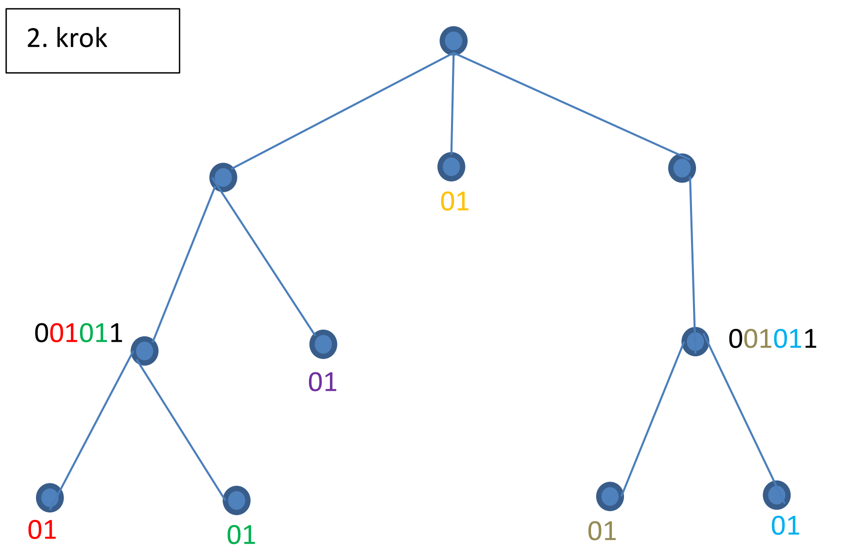


Nejdřív každý list dostane kód



Pak budeme přiřazovat kódy ostatním vrcholům. Postupujeme od spodních vrstev, nejspodnější už je. Přímý předchůdce zdědí kódy svých přímých potomků **uspořádané lexikografickým pravidlem**, a pak připíšeme na začátek 0 a na konec 1. Lexikografické pravidlo teď znamená, kódy porovnáváme po cifrách:

V tomto kroku srovnáváme stejné kódy (01, 01), jsou stejné, na pořadí nezáleží a píšeme je zleva doprava jako u klasického kódu v předchozí úloze:

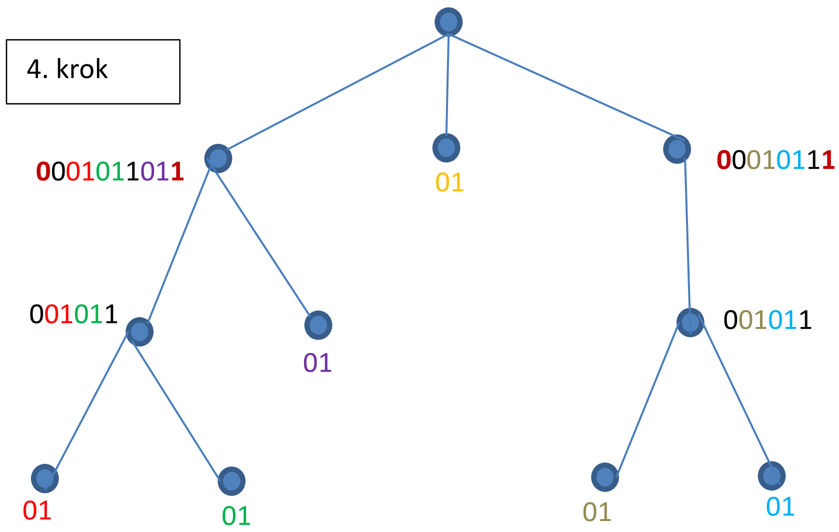


V dalším kroku budeme srovnávat kódy:

0 0 1 0 1 1

0 1

Můžeme vidět, že když srovnáme první cifry, jsou stejné (0=0), pak srovnáme druhé cifry: 0 < 1, to znamená, že kód 0 0 1 0 1 1 < 0 1, proto i když je kód delší, je lexikograficky menší a bude první.



V dalším kroku srovnávamé kódy:

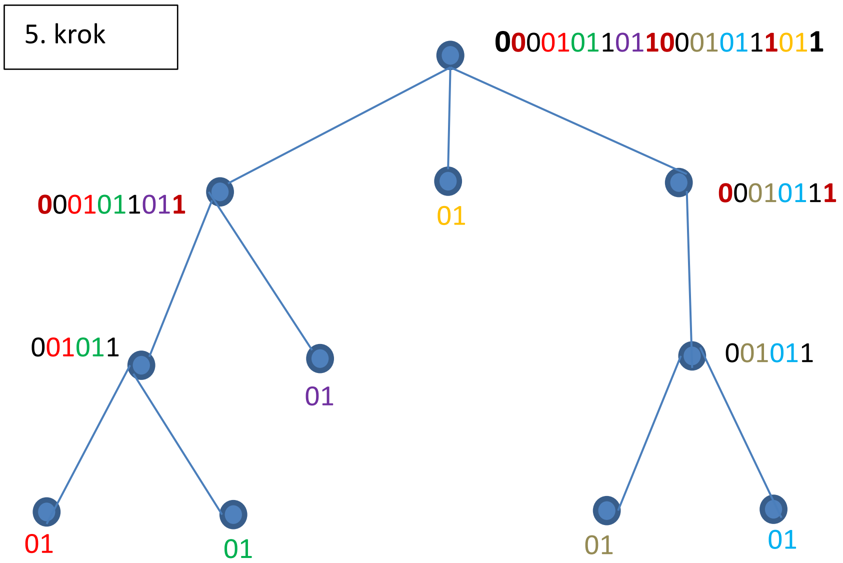
0 0 0 1 0 1 1 0 1 1

0 1

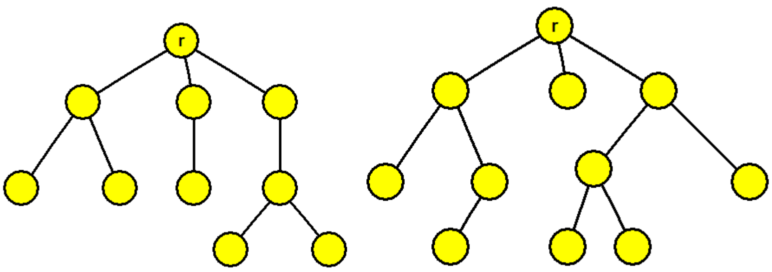
0 0 0 1 0 1 1 1

První srovnání 0 = 0 = 0, zatím všechny stejné. Druhé srovnání 0 < 1 > 0, kód 01 je největší a bude poslední. Dál pokračujeme v porovnávání dalších dvou kódů: 0 = 0, 1 = 1, 0 = 0, 1 = 1, 1 = 1, 0 < **1**, teda kód **0** 0 0 1 0 1 1 0 1 **1** < **0** 0 0 1 0 1 1 **1.** Máme teda srovnání všech třech kódu:

- **0** 0 0 1 0 1 1 0 1 **1** < **0** 0 0 1 0 1 1 **1** < 0 1 a v tomto pořadí je zdědí přímý předchůdce:



Analogicky vytvořte minimální kódy následujícím stromům.



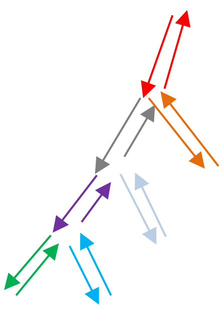
Úloha 8: Nakreslete stromy

K následujícím kódům nakreslete kořenové stromy:

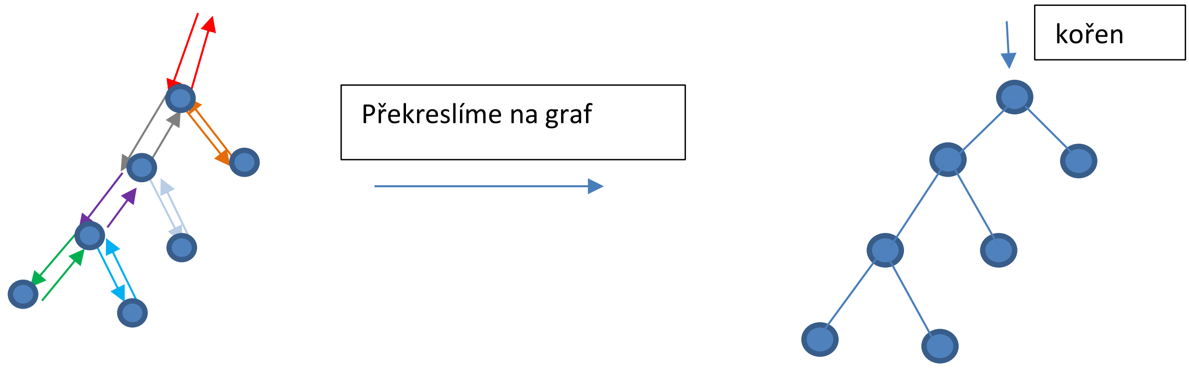
- **0 0 0 0 1 0 1 1 0 1 1 0 1 1**

Řešení:

0 znamená šipku dolů a 1 šipku, která se vrací zpátky po 0.



Teď všude mezi šipky a konce vložíme vrchol, kromě první červené – je to ukazovatel na kořen.



Analogicky udělejte následující:

- 0001001011100101111
- 00100100101111

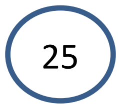
Úloha 9: Binární vyhledávací strom

- Pomocí operace *insert* vložte čísla 25, 1, 18, 5, 31, 14, 30, 31, 4, 40, 60 do binárního stromu a následně odstraňte kořen.

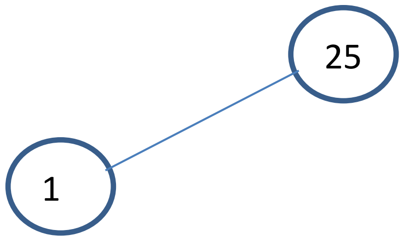
Řešení:

Postupně budeme vytvářet binární strom tak, že když vkládáme číslo s menší hodnotou, vložíme ho vlevo, když s větší hodnotou, tak ho vložíme vpravo:

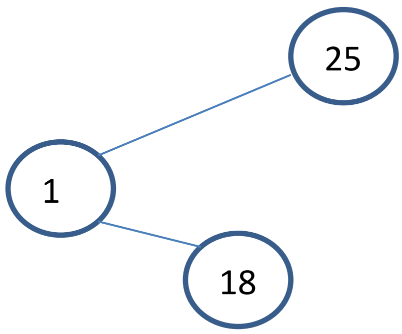
25– první prvek bude kořen



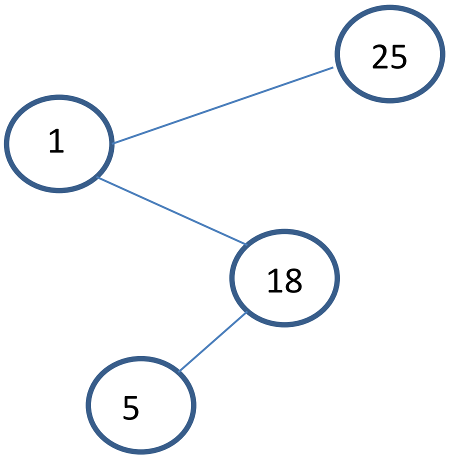
1– druhý prvek je menší než 25, vložíme ho vlevo



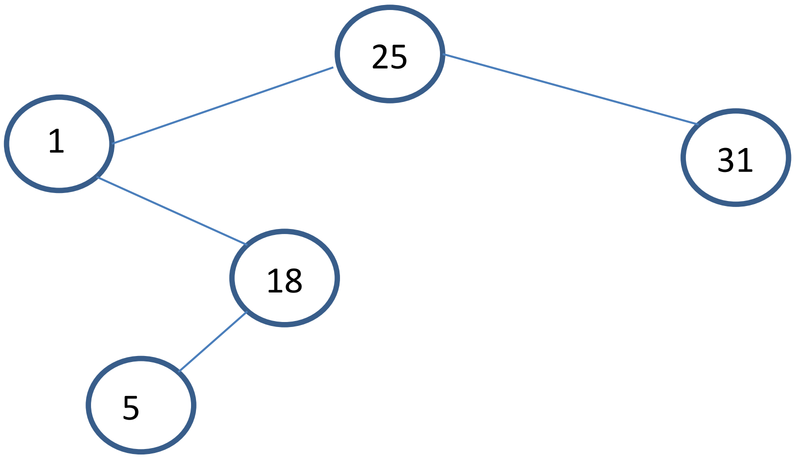
18-třetí prvek je menší než 25, jdeme vlevo, je větší než 1, vložíme vpravo:



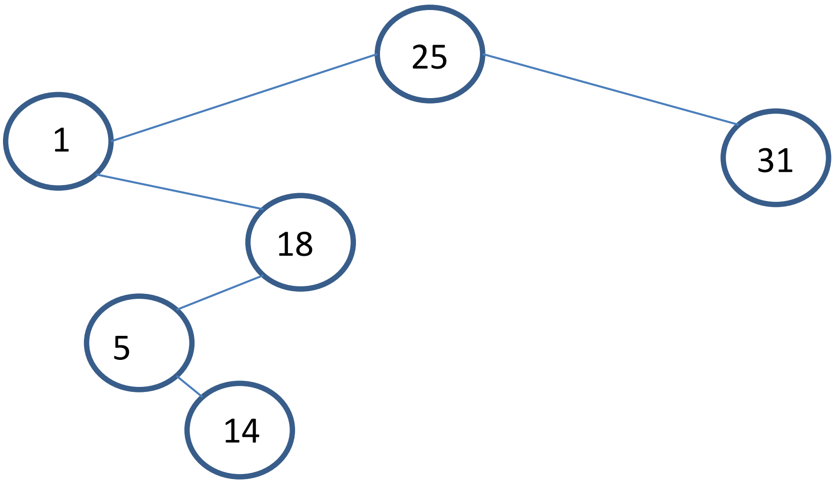
5-čtvrtý prvek je menší než 25, jdeme vlevo, větší než 1, jdeme vpravo, menší než 18, vložíme ho vlevo:



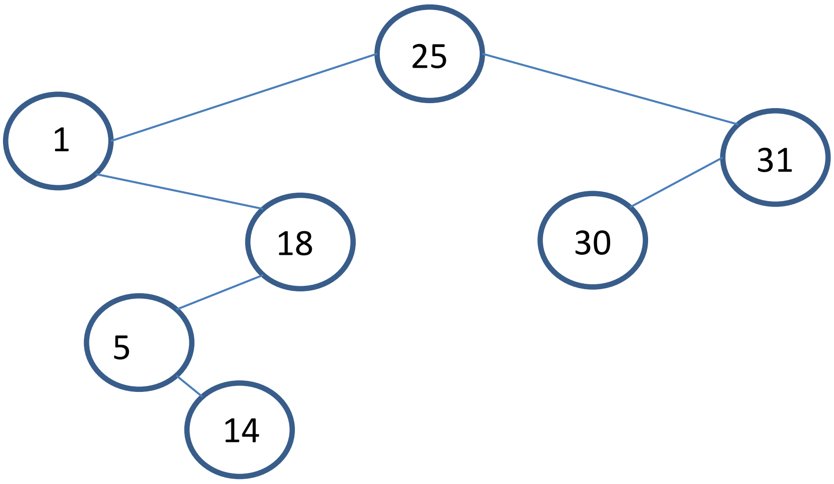
31-pátý prvek je větší než 25, zapíšeme vpravo:



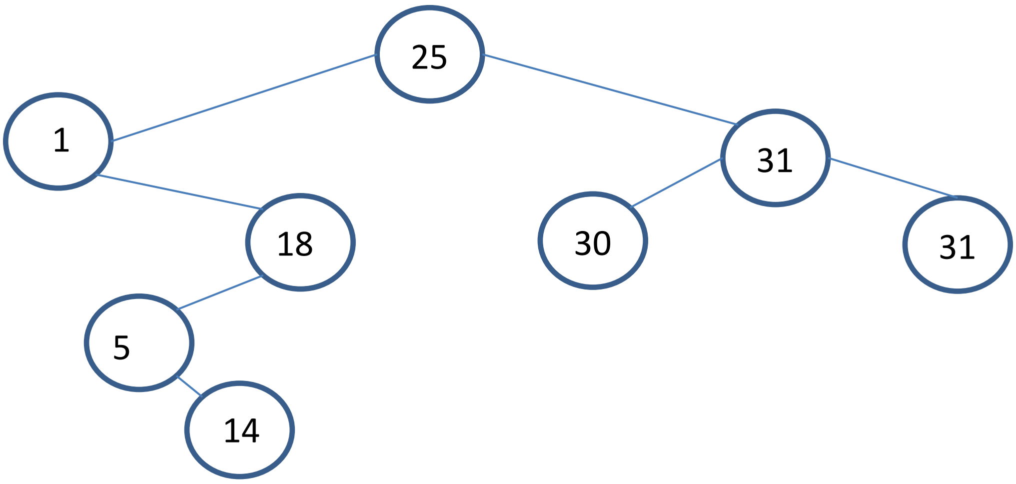
14-šestý prvek je menší než 25 → vlevo, je větší než 1 → vpravo, menší než 18 → vlevo, větší než 5 → zapíšeme vpravo:



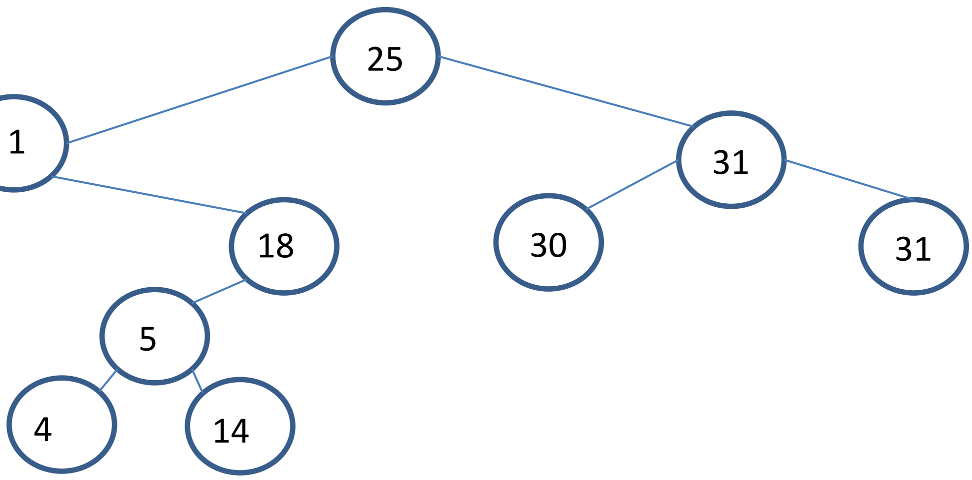
30-sedmý prvek je větší než 25 → vpravo, je menší než 31, zapíšeme vlevo:



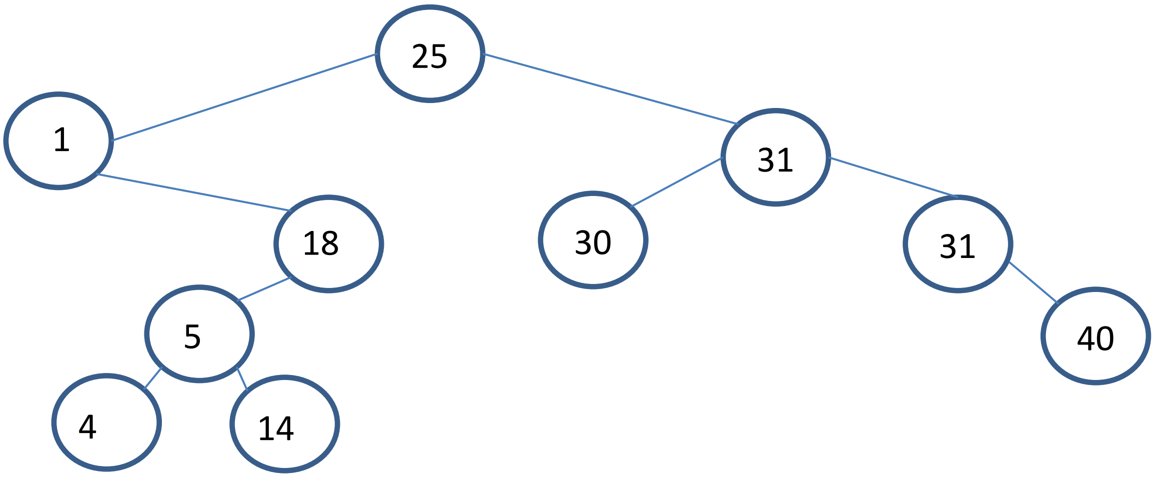
31- osmý prvek je větší než 25 → vpravo, je stejný jako 31, můžeme i vlevo i vpravo, např. vpravo nic není, tak ho tam můžeme zapsat vpravo:



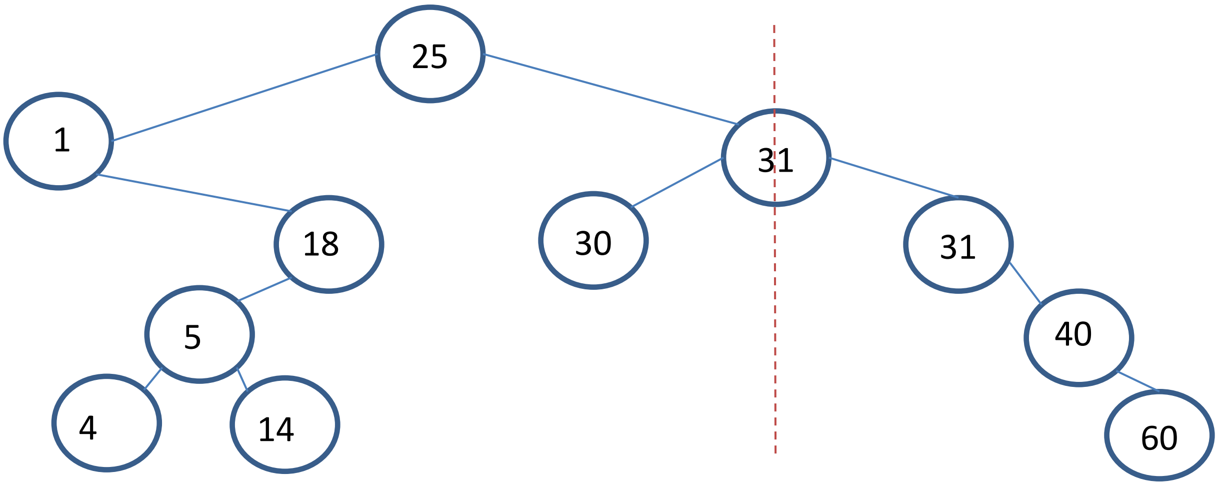
4 – devátý prvek je menší než 25 → vlevo, větší než 1 → vpravo, menší než 18 → vlevo, menší než 5 → zapíšeme vlevo



40- desátý prvek je větší než 25 → vpravo, větší než 31 → vpravo, větší než 31 → vložíme vpravo:

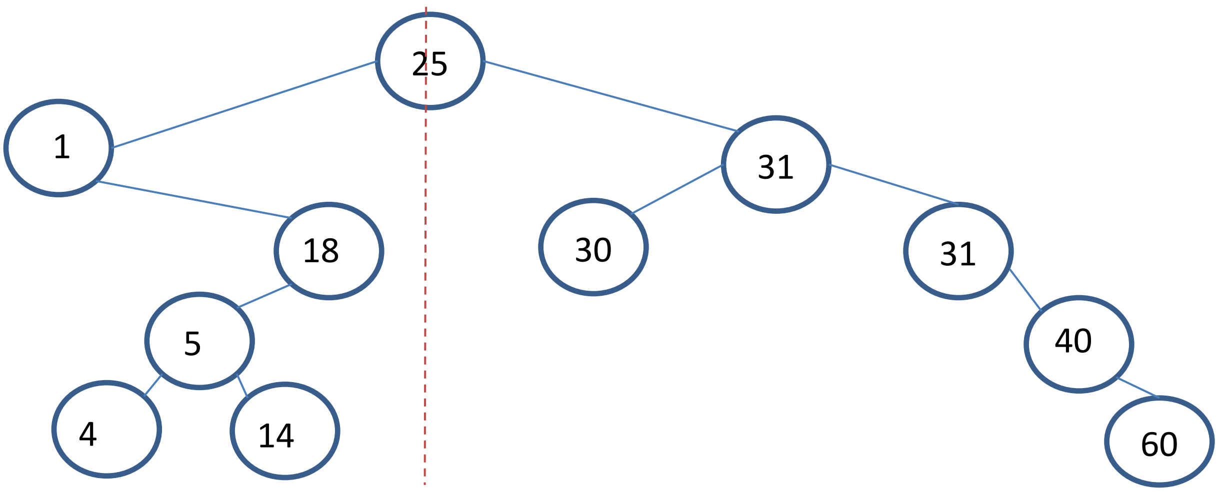


60-jedenáctý prvek je větší než 25 → vpravo, větší než 31 → vpravo, větší než 31 → vpravo, větší než 40 → zapíšeme vpravo:

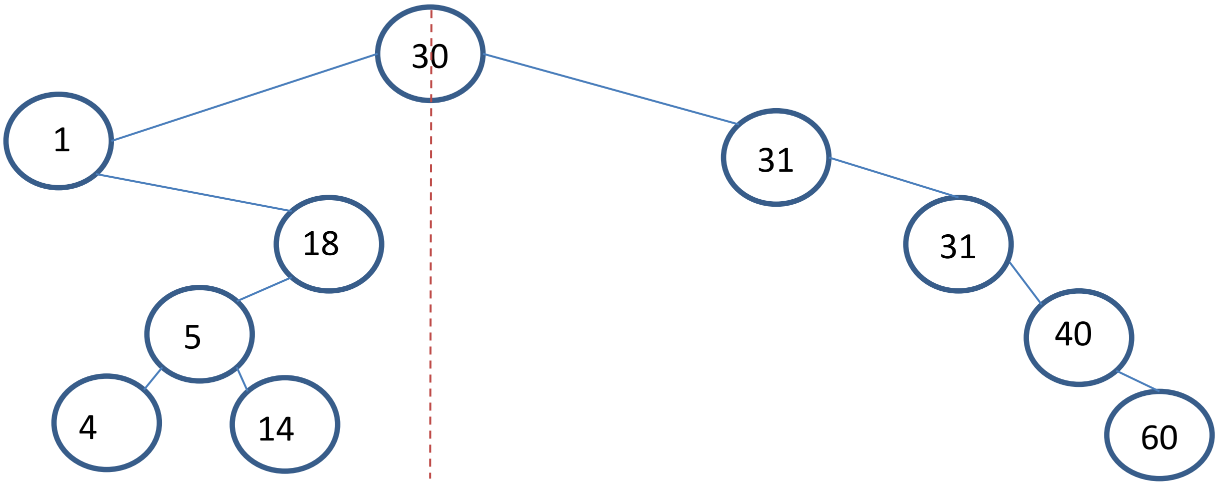


Strom je potřeba kreslit tak, že vždy vlevo jsou menší/rovné prvky než prvek, přes který byste přeložili čáru a vpravo větší/rovné.

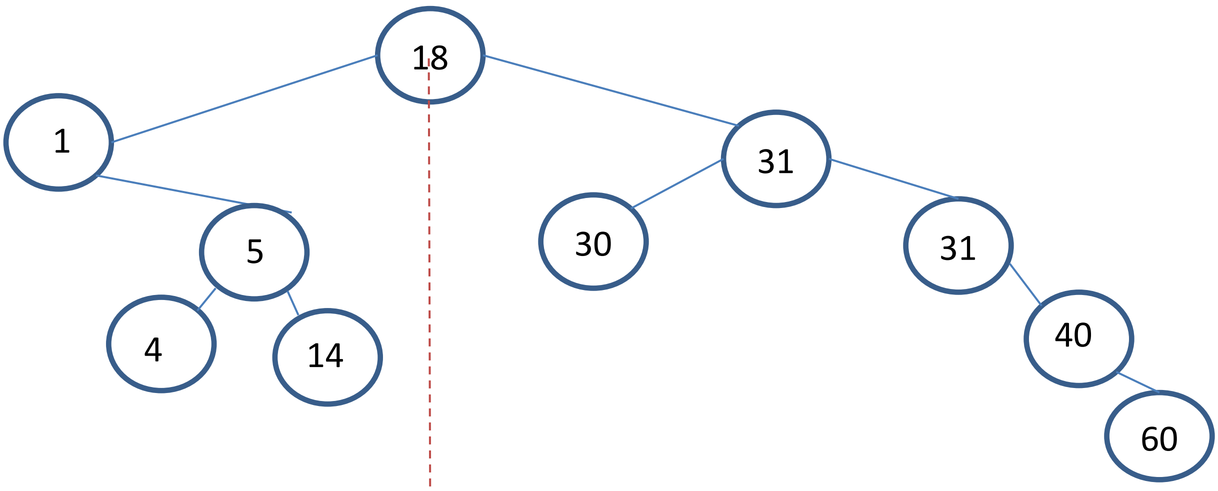
Ještě odebereme kořen:



A nahradíme ho buď nejlevějším prvkem z pravé části, to je 30:



Nebo nejpravějším prvkem z levé částí, to je 18:



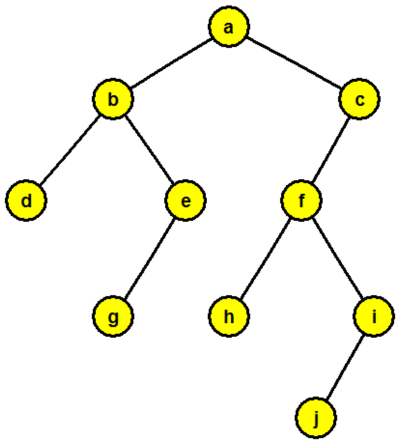
Analogicky udělejte zbytek:

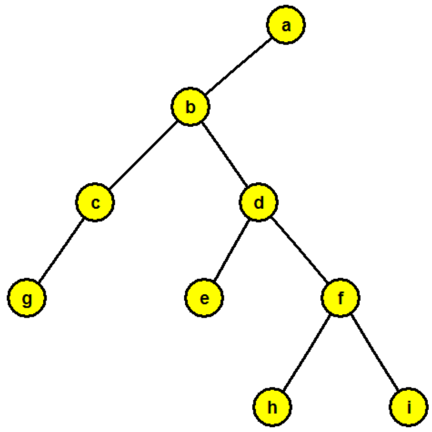
- Pomocí operace *insert* vložte čísla 2, 23, 4, 76, 23, 3, 54, 311, 14, 22, 1 do binárního stromu a následně odstraňte kořen.
- Pomocí operace *insert* vložte čísla 54, 22, 11, 44, 55, 77, 34, 2 do binárního stromu a následně odstraňte kořen.

Úloha 10: Prohledávání binárních stromů

V následujících stromech zpracujte vrcholy a následně vypište pomocí procedury Preorder, Inorder a Postorder.

Procedury jsou popsané v prezentaci a v učebních textech. Pokuste se podle nich zpracovat vrcholy v následujících binárních stromech:





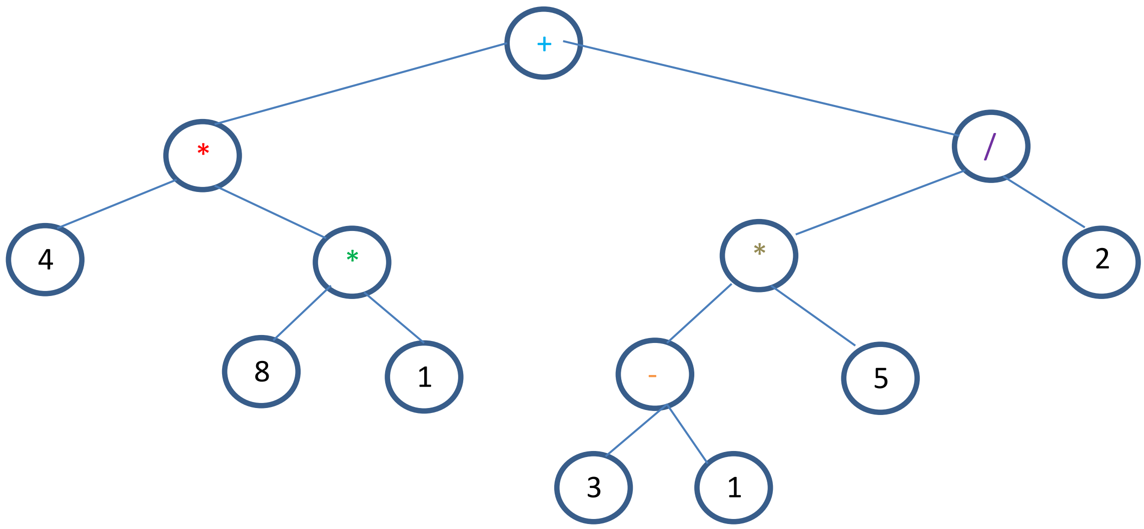
Úloha 11: Prohledávání binárních stromů

Pomocí vhodné procedury řešte výraz:

- (4*(8*1))+(((3-1)*5)/2)

Řešení:

Nejdřív výraz vložíme do binárního stromu:



Provedeme proceduru INORDER: +,*,4,*,8,1,/,*,-,3,1,5,2

Najdeme operaci i hned dvě následující čísla a provedeme operaci mezi nimi:

+,*,4,*,8,1,/,*,-,3,1,5,2

+,*,4, 8, /,*, 2, 5,2

+, 32, /, 10, 2

+, 32, 5

37

Výraz (4*(8*1))+(((3-1)*5)/2) je roven 37.

Analogicky proveďte následující úkol:

- (12/(3*(4-2)))*(12/6)/3)

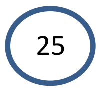
Úloha 11: Halda

- Pomocí operace *insert* vložte čísla 25, 1, 18, 5, 31, 14, 30, 31, 4, 40, 60 do haldy a následně odeberte kořen.

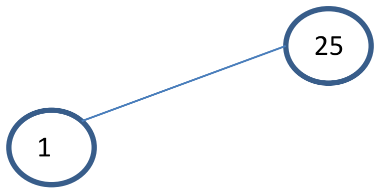
Řešení:

Podle návodu z prezentace/učebního textu v olivě sestrojíme haldu:

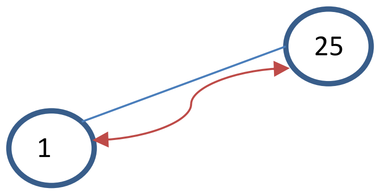
25, 1, 18, 5, 31, 14, 30, 31, 4, 40, 60

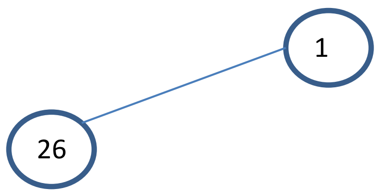


25, 1, 18, 5, 31, 14, 30, 31, 4, 40, 60

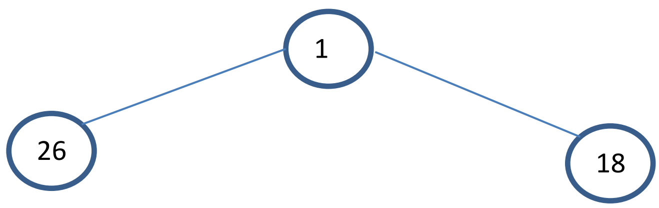


Není splněna podmínka haldy (menší prvek je vždy víš), uděláme prohození:

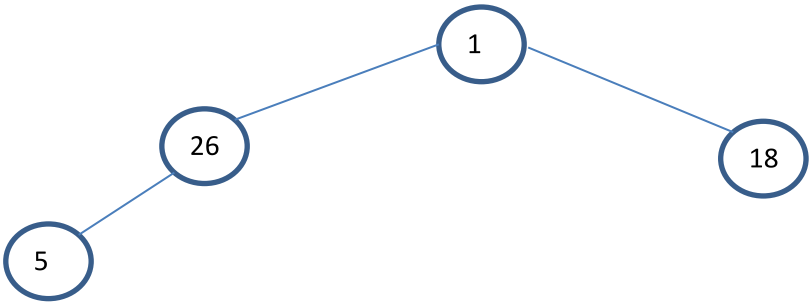




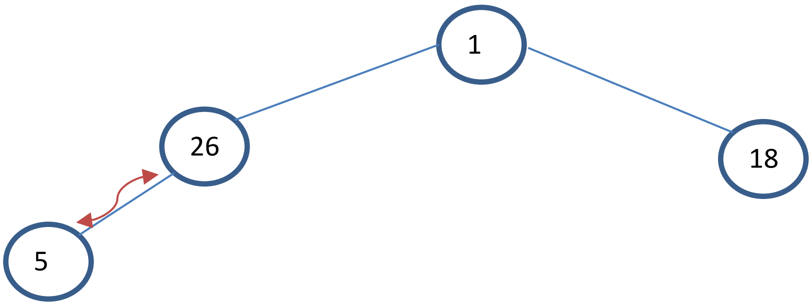
Vložíme další prvek, 25, 1, 18, 5, 31, 14, 30, 31, 4, 40, 60:

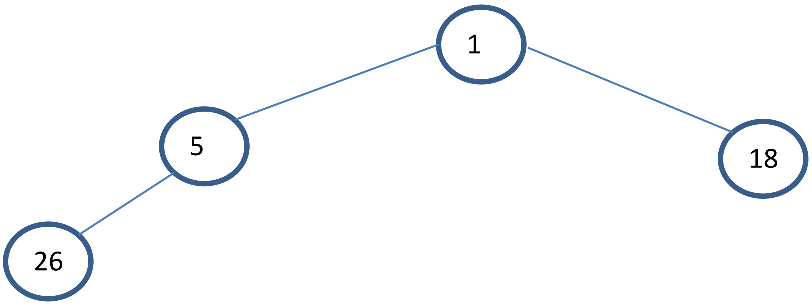


Vlastnost haldy je ok, vkládáme další prvek, 25, 1, 18, 5, 31, 14, 30, 31, 4, 40, 60:

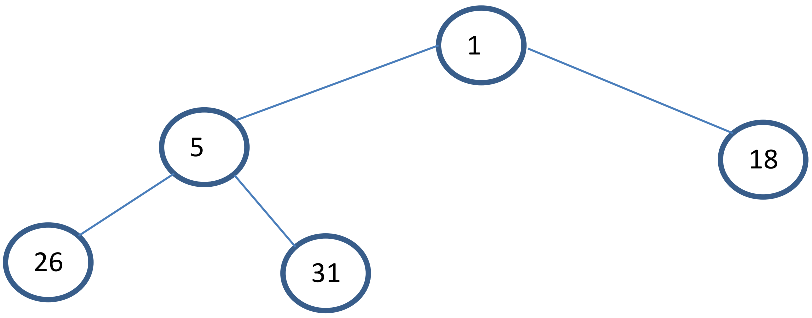


Vlastnost haldy splněná není, budeme prohazovat, abychom dostali menší prvek výše:

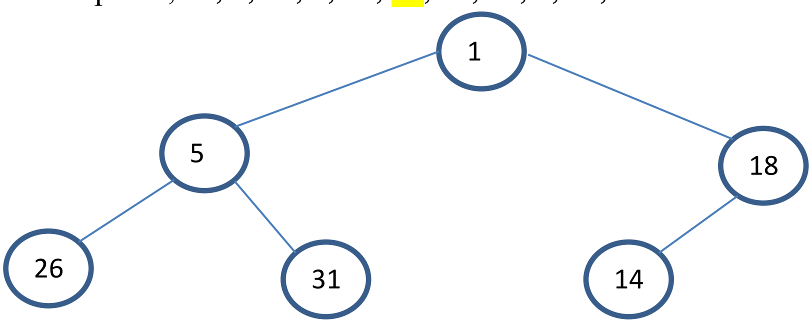




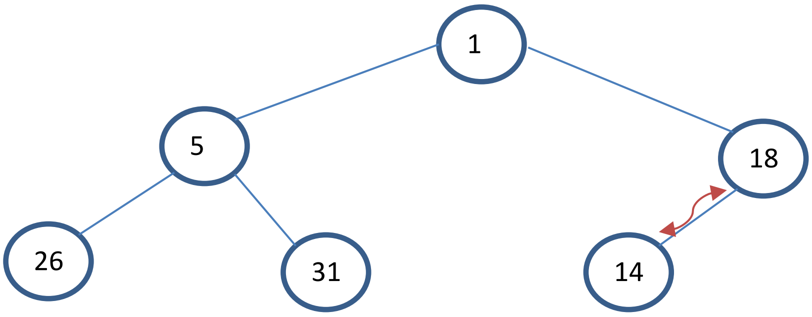
Vložíme další prvek, 25, 1, 18, 5, 31, 14, 30, 31, 4, 40, 60:

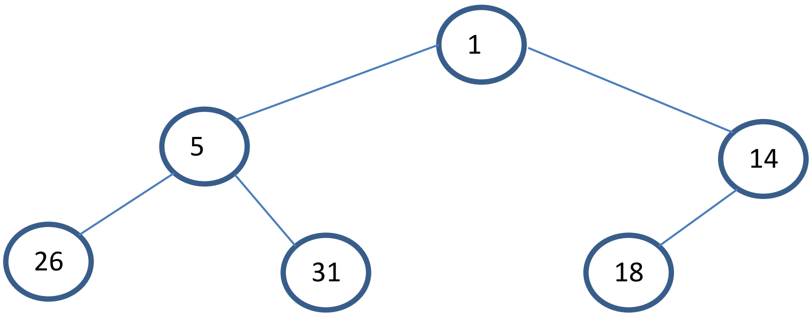


Vložíme další prvek, 25, 1, 18, 5, 31, 14, 30, 31, 4, 40, 60:

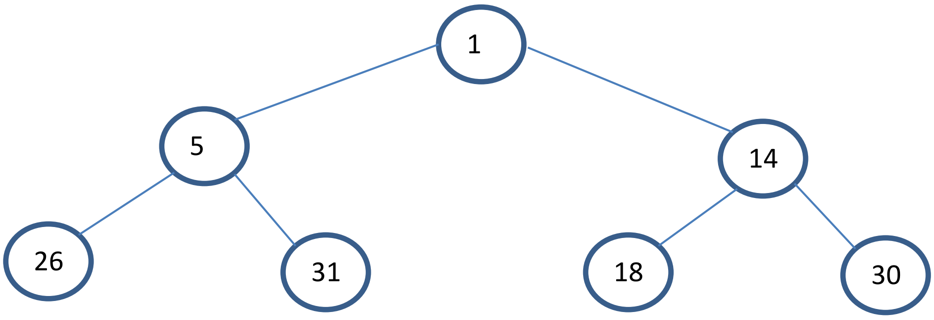


Není splněná vlastnost haldy, budeme prohazovat:

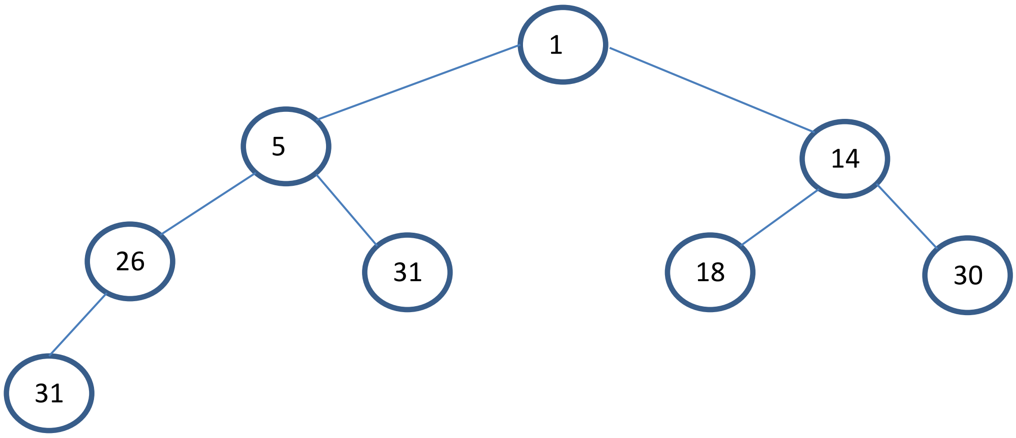




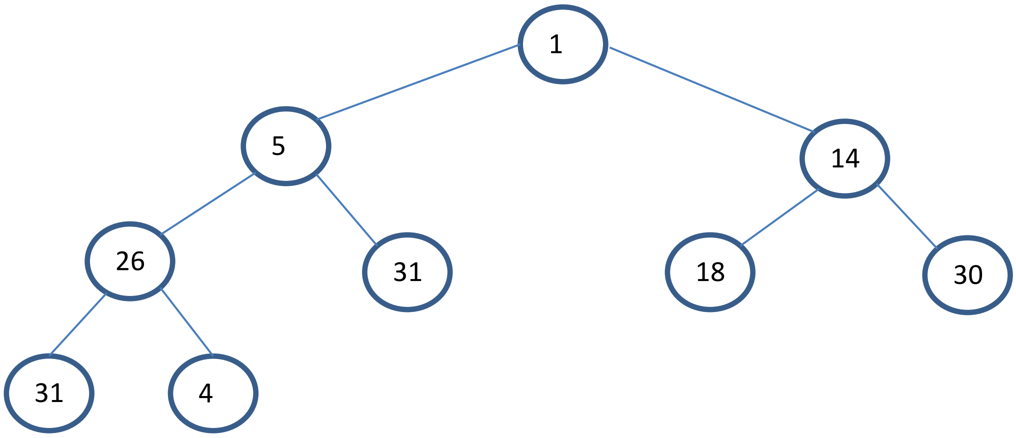
Vložíme další prvek, 25, 1, 18, 5, 31, 14, 30, 31, 4, 40, 60:



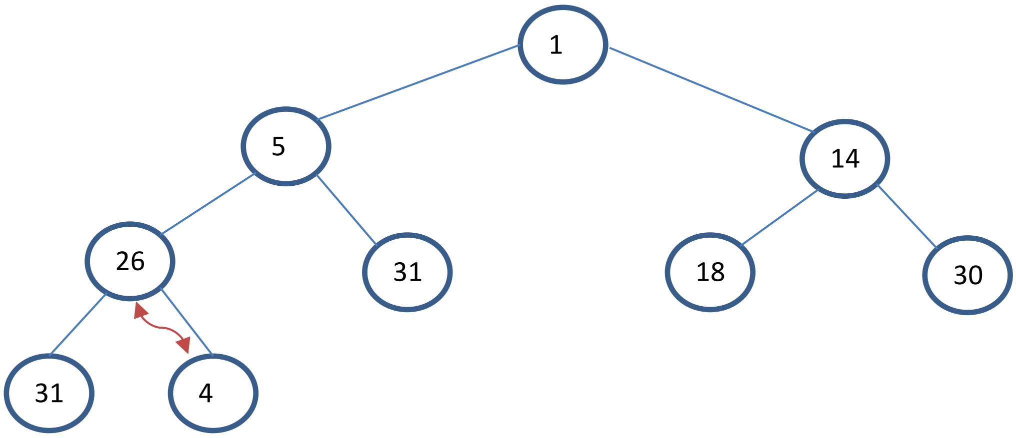
Vložíme další prvek, 25, 1, 18, 5, 31, 14, 30, 31, 4, 40, 60:

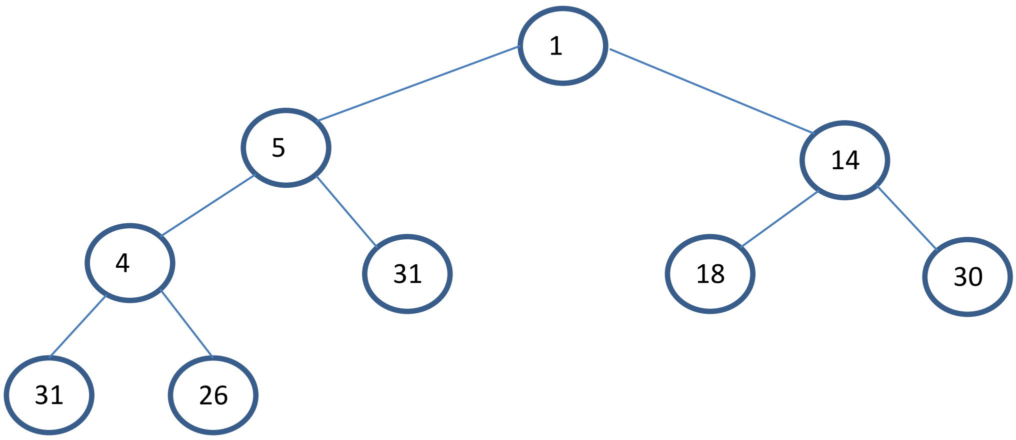


Vložíme další prvek, 25, 1, 18, 5, 31, 14, 30, 31, 4, 40, 60:

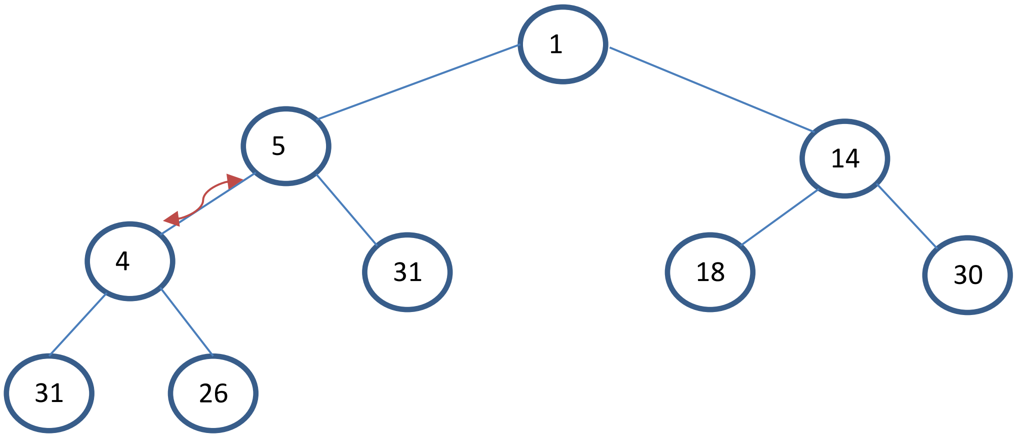


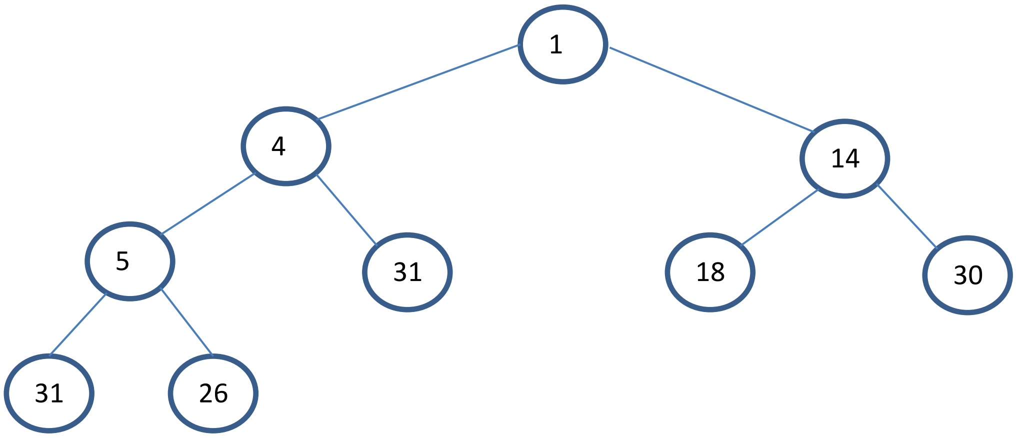
Nesplněná vlastnost haldy, budeme prohazovat:



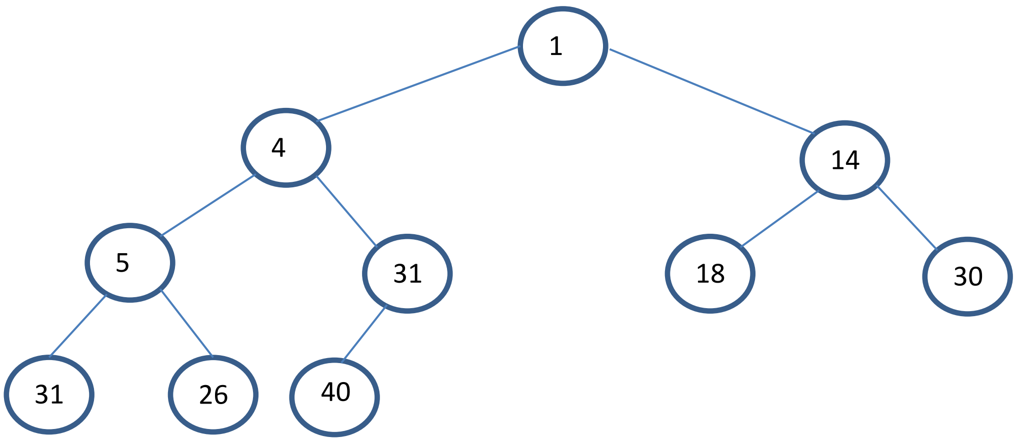


Ještě není splněná vlastnost, prohazujeme dál:

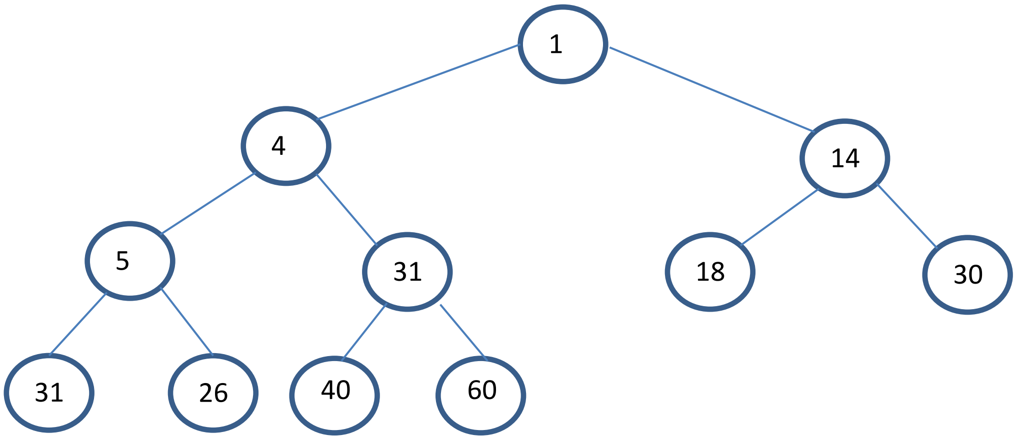




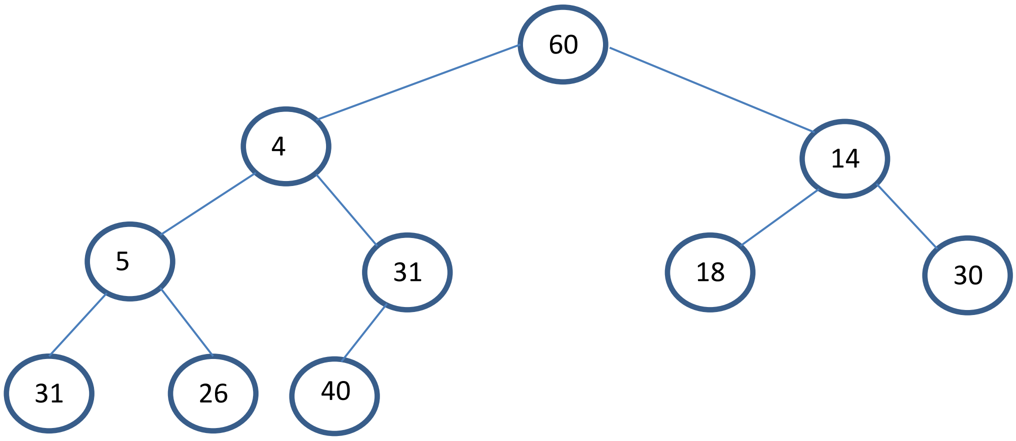
Vložíme další prvek, 25, 1, 18, 5, 31, 14, 30, 31, 4, 40, 60:



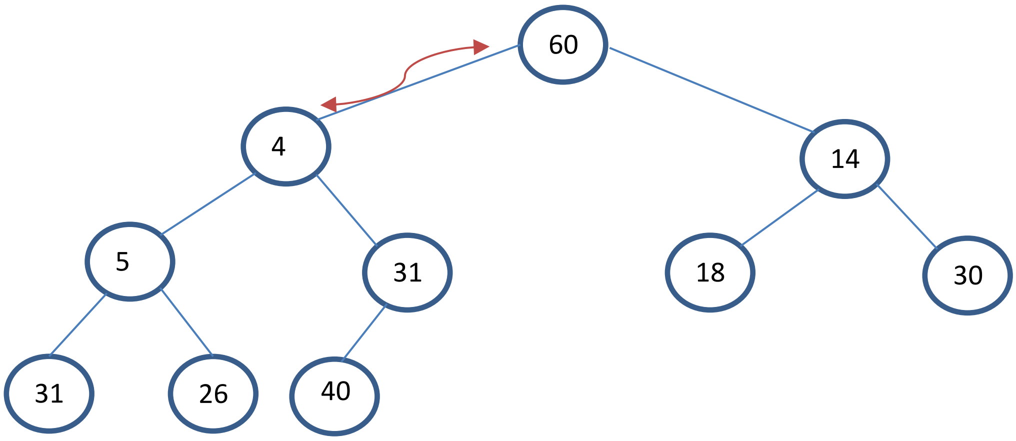
Vložíme další prvek, 25, 1, 18, 5, 31, 14, 30, 31, 4, 40, 60:

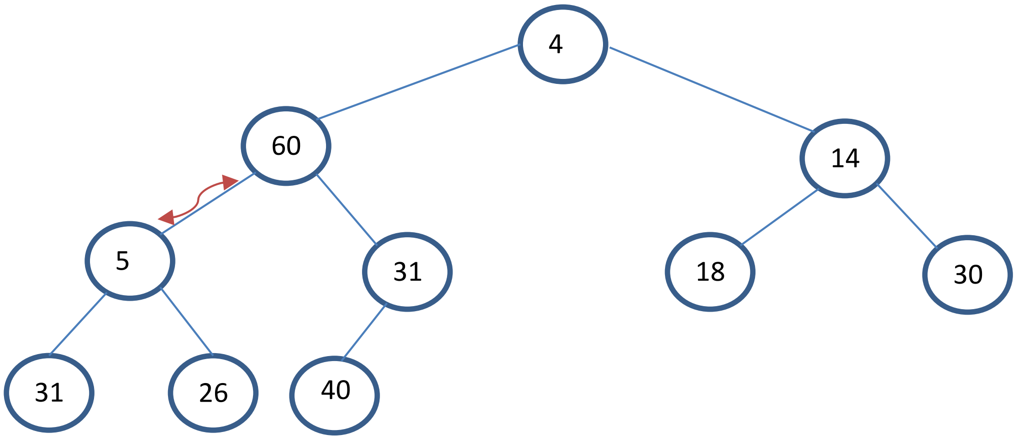


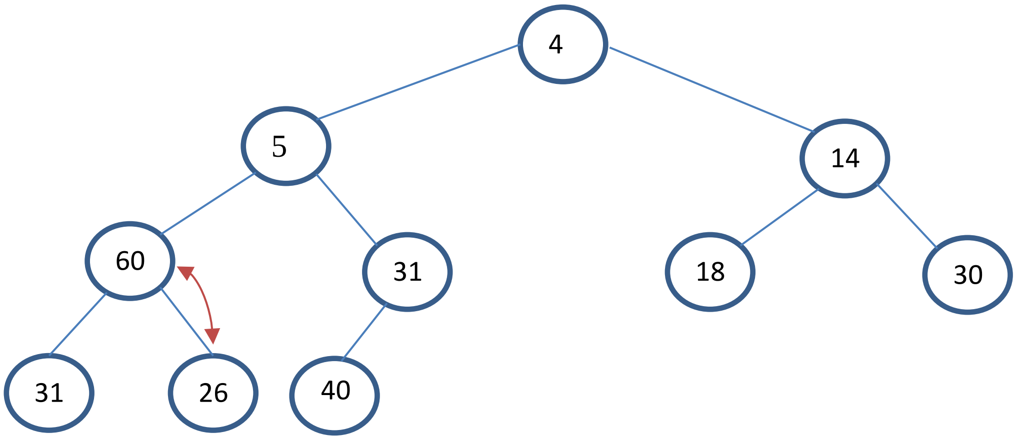
Halda je vytvořená a v kořeni je nejmenší prvek, teď ho odebereme a nahradíme ho posledním prvkem v haldě a to 60:

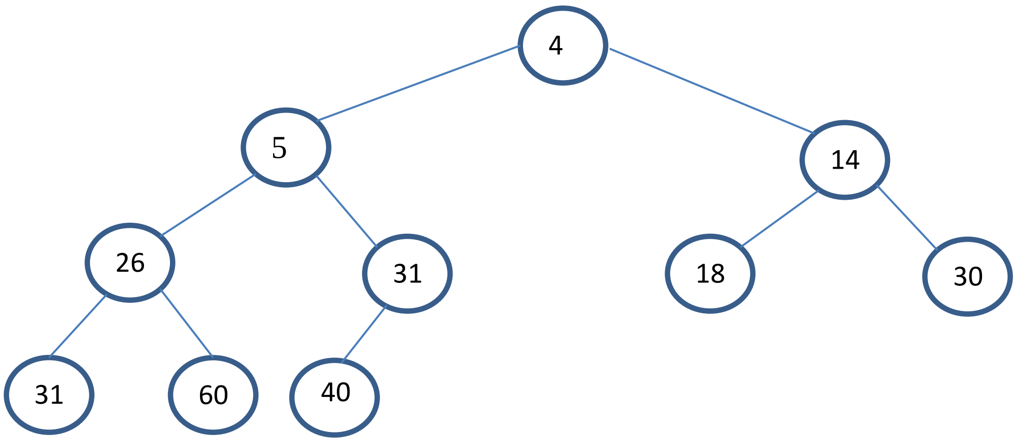


Byla porušená vlastnost, proto budeme prohazovat (prohazujeme vždy s menším prvkem), dokud nedosáhneme splnění vlastností, že nahoru je vždy menší prvek:









Vlastnost je splněná.

Analogicky řešte následující úlohy:

- Pomocí operace *insert* vložte čísla 2, 23, 4, 76, 23, 3, 54, 311, 14, 22, 1 do haldy a následně odeberte minimum.
- Pomocí operace *insert* vložte čísla 54, 22, 11, 44, 55, 77, 34, 2 do haldy a určete maximum.

Úloha 12: Důkazy pro zájemce

Pokuste se nejdřív samostatně (důkazy pak sepíšu všechny spolu a pošlu):

- Dokažte přímo tvrzení: Jestliže graf *G* je strom, pak graf *G* – *e*, kde *e* je libovolná hrana grafu *G*, již není strom. (K důkazu použijte známé definice a věty týkající se stromů.)

- Dokažte sporem větu: Jestliže graf *G* je strom, pak graf *G* není hamiltonovský. (Nejprve správně zformulujte tvrzení pro důkaz sporem!)

- Dokažte pomocí nepřímého důkazu tvrzení: Jestliže každá hrana grafu je most, pak graf je les. (Nejprve správně zformulujte tvrzení pro nepřímý důkaz!)