ADSO
7. Stromy

(prezentace k učebnici)

Ivan Ryant

Agenda

- Základní pojmy
- Úloha abstraktních datových typů
- Posloupnosti a operace s nimi
- Vyhledávací datové struktury
- Vyhledávací posloupnost
- Algoritmy řazení
- **Stromy**
- Rozptýlené tabulky
- Prohledávání do hloubky a do šířky
- Práce s grafy
- Techniky návrhu efektivních algoritmů

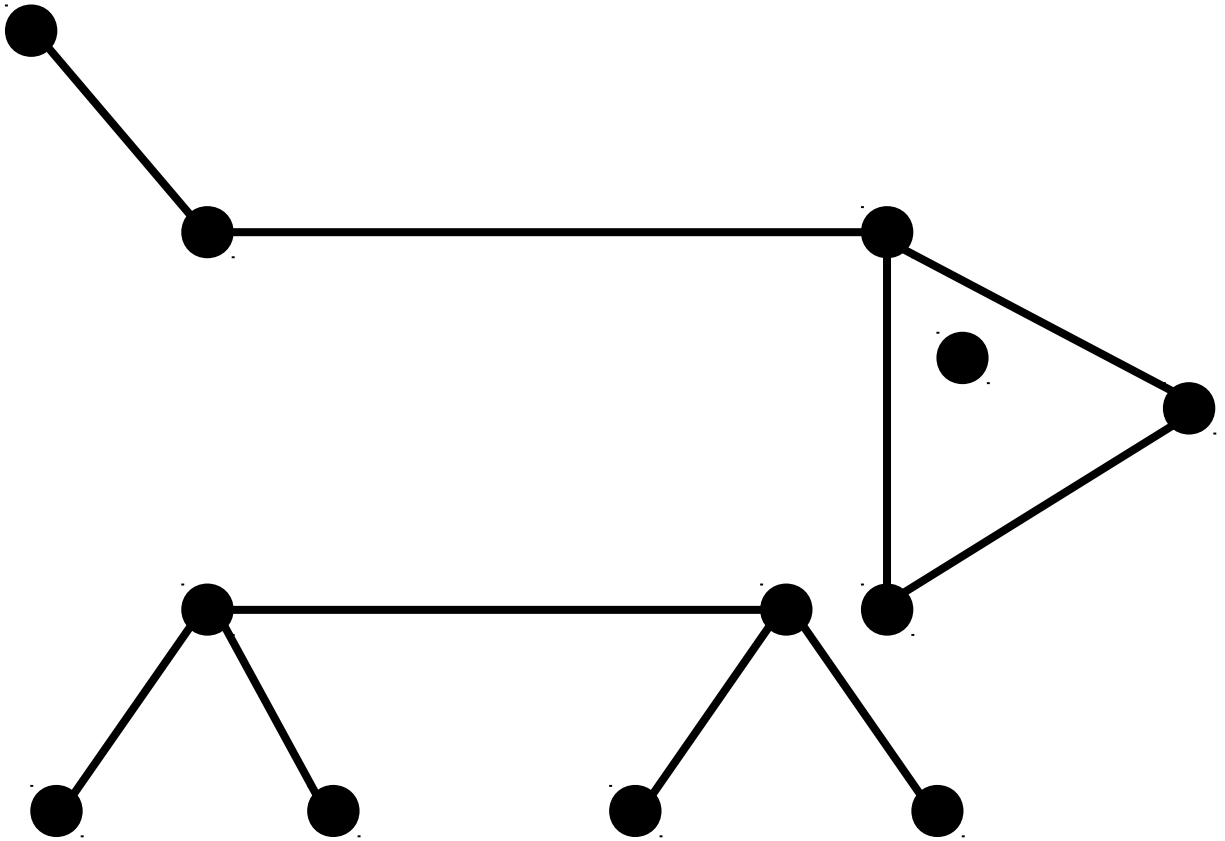
Stromy

- Agenda
  - **Pojmy graf a strom**
  - Halda
  - Binární vyhledávací strom
  - Bisekce
  - Vyhledávání v binárním vyhledávacím stromě
  - Vyvážování binárních vyhledávacích stromů
  - Shrnutí

Graf a strom

- Graf je algebraická struktura, rozlišujeme
  - vrcholy
  - hrany
  - incidenční funkci
- Graf může být
  - orientovaný
    - každá hrana rozlišuje mezi výchozím a cílovým vrcholem
  - neorientovaný
    - hrany nerozlišují, který vrchol je výchozí nebo cílový
- Strom je graf, který má specifické vlastnosti

Graf a strom



Příklad nakreslení grafu
(3 komponenty souvislosti, 1 cyklus, mnoho cest)

Graf a strom

- Orientovaný graf je uspořádaná trojice
  $G = \langle V, H, i \rangle$
  složená postupně ze tří součástí:
  - množina vrcholů (někdy také uzlů) V
  - množina hran H
  - incidenční funkce $i : H \rightarrow \langle u, v \rangle$, kde $u, v \in V$

    - Všimněte si, že $\langle u, v \rangle$ je **uspořádaná** dvojice, která umožňuje rozlišit vrchol výchozí od vrcholu cílového

Graf a strom

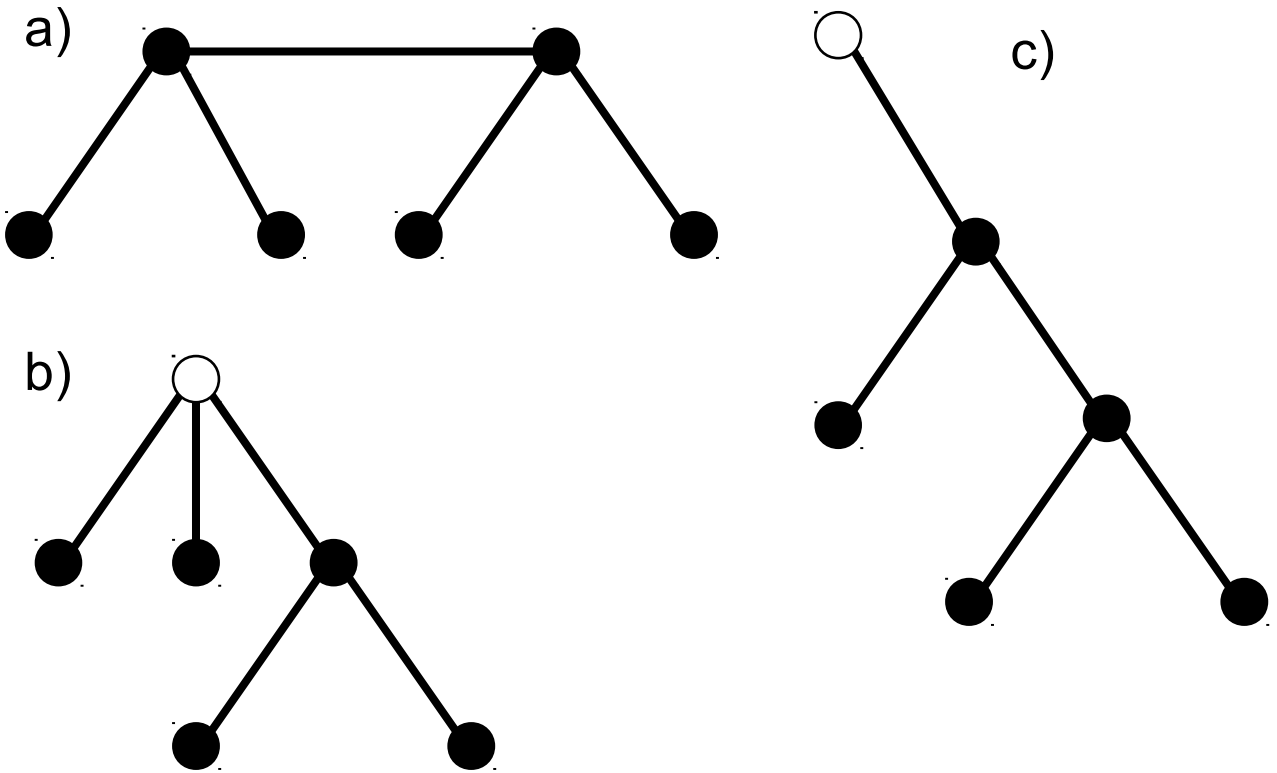
- Neorientovaný graf je uspořádaná trojice $G = \langle V, H, i \rangle$
  složená postupně ze tří součástí:
  - množina vrcholů (někdy také uzlů) V
  - množina hran H
  - incidenční funkce $i : H \rightarrow \{u, v\}$, kde $u, v \in V$

    - Všimněte si, že {u, v} je **neuspořádaná** dvojice, která neumožňuje rozlišit vrchol výchozí od vrcholu cílového

Graf a strom

- **Cesta** v grafu je posloupnost vrcholů
  - Z kteréhokoli vrcholu cesty existuje hrana do jeho následníka
    - kromě posledního vrcholu (ten nemá následníka)
  - Žádná dvojice vrcholů (a tedy ani hrana) se přitom neopakuje.
- O grafu říkáme, že je **souvislý,** když
  - každá dvojice vrcholů je v něm spojena cestou.
  - A naopak: jestliže některé dva vrcholy nejsou spojeny cestou, graf není souvislý. Jestliže graf není souvislý, tak jeho souvislé podgrafy nazýváme komponenty souvislosti.
- **Cyklus** (neboli kružnice) v grafu
  - je posloupnost vrcholů podobná cestě. Od cesty se liší tím, že
  - všechny vrcholy v cyklu mají jak jednoho předchůdce, tak jednoho následníka.
  - Jestliže je cyklus tvořen jediným vrcholem, který je spojen sám se sebou jedinou hranou, nazývá se **smyčka.**
  - Graf je **acyklický,** když neobsahuje žádný cyklus.
- **Strom** je souvislý acyklický neorientovaný graf.

Graf a strom



Tři nakreslení téhož stromu

a) kořen není zvolen
b) kořen je zvolen tak, že strom není binární (je ternární) a má 4 listy
c) binární strom se třemi listy

Stromy

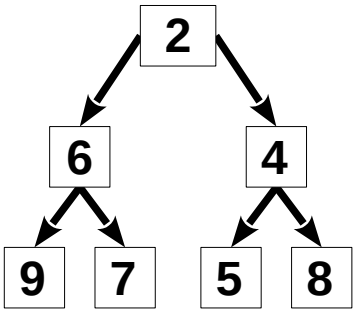
- Agenda
  - Pojmy graf a strom
  - **Halda**
  - Binární vyhledávací strom
  - Bisekce
  - Vyhledávání v binárním vyhledávacím stromě
  - Vyvážování binárních vyhledávacích stromů
  - Shrnutí

Halda

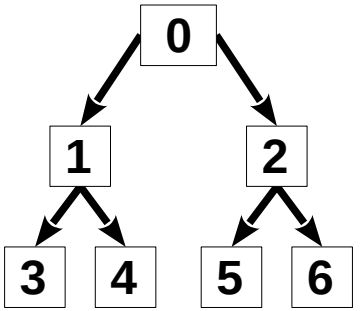
Příklad: Máme posloupnost prvků s klíči:

8 9 4 7 2 5 6

a chceme ji uspořádat do haldy. Halda bude uspořádaná třeba takto:



Vrcholům haldy přiřadíme indexy:



Haldu umístíme do pole takto:

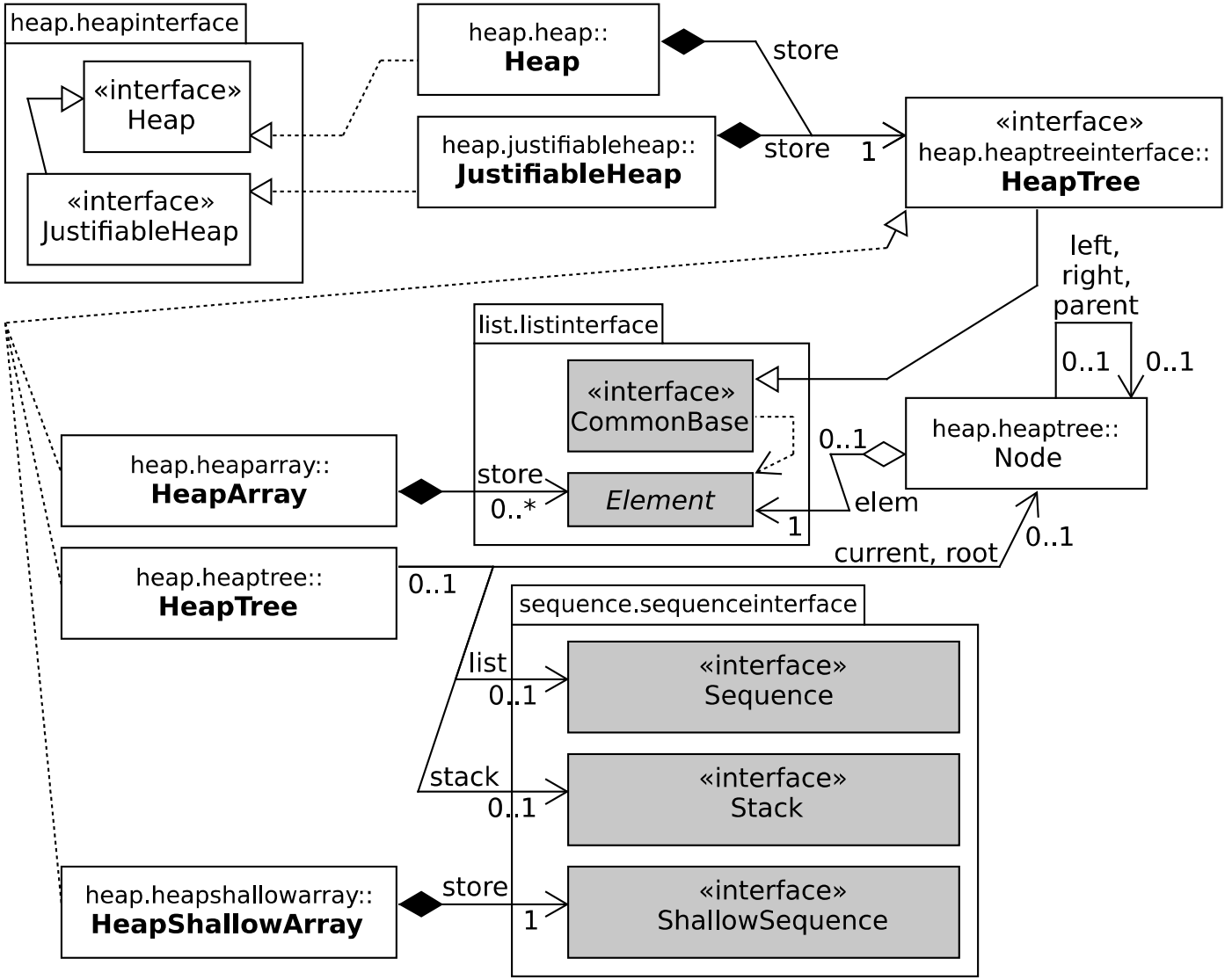
0	1	2	3	4	5	6
2	6	4	9	7	5	8

Příklad haldy a jejího umístění do pole

Halda

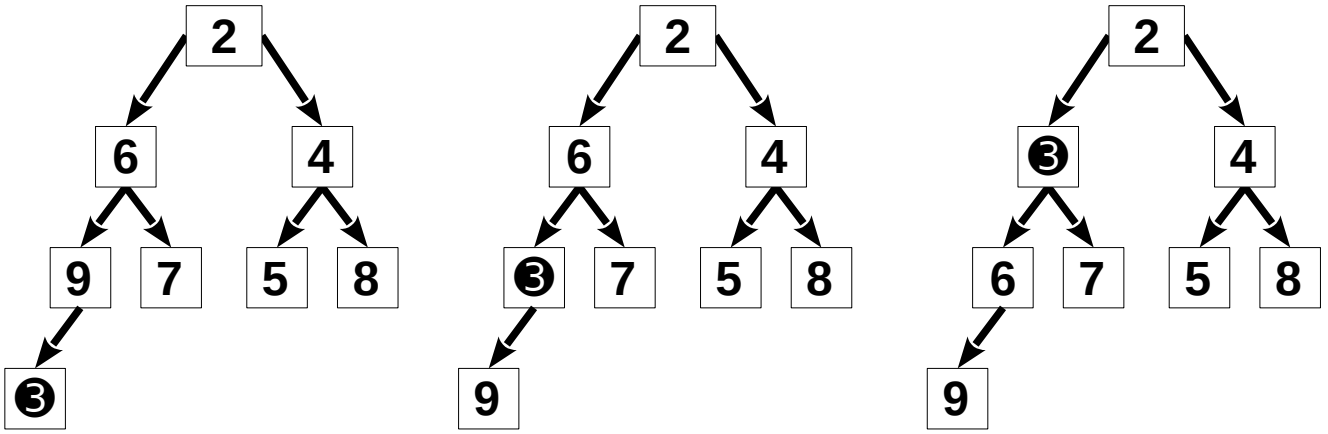
- **Halda** je binární strom
  s těmito zvláštními vlastnostmi:
  - Předek má menší klíč než kterýkoli z jeho potomků.
  - Všechny vrstvy kromě poslední musí být plné.
  - Poslední vrstva se zaplňuje zleva doprava (bez vynechávání).

Halda

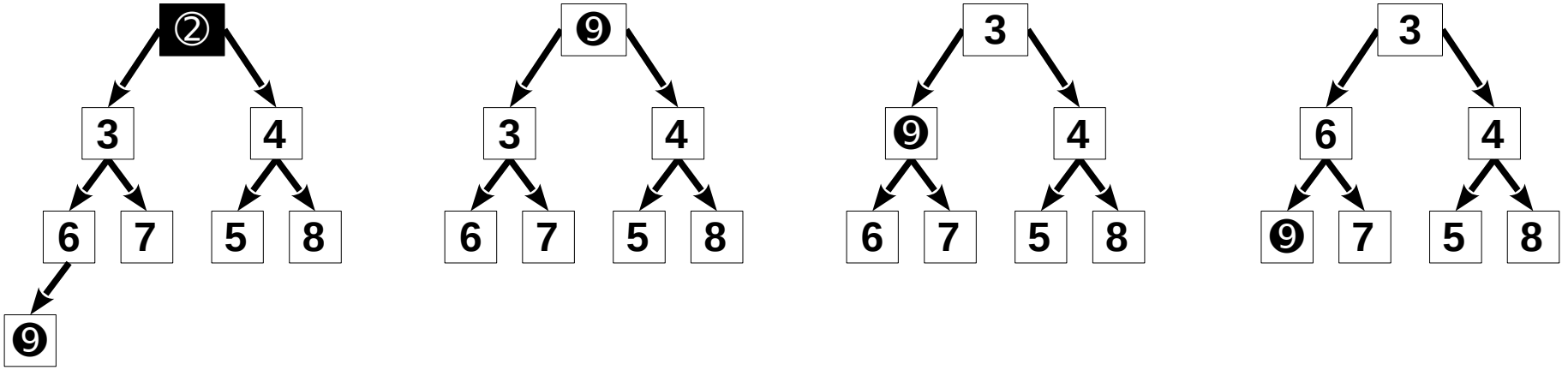


Design haldy

Halda



Příklad přidání prvku do haldy

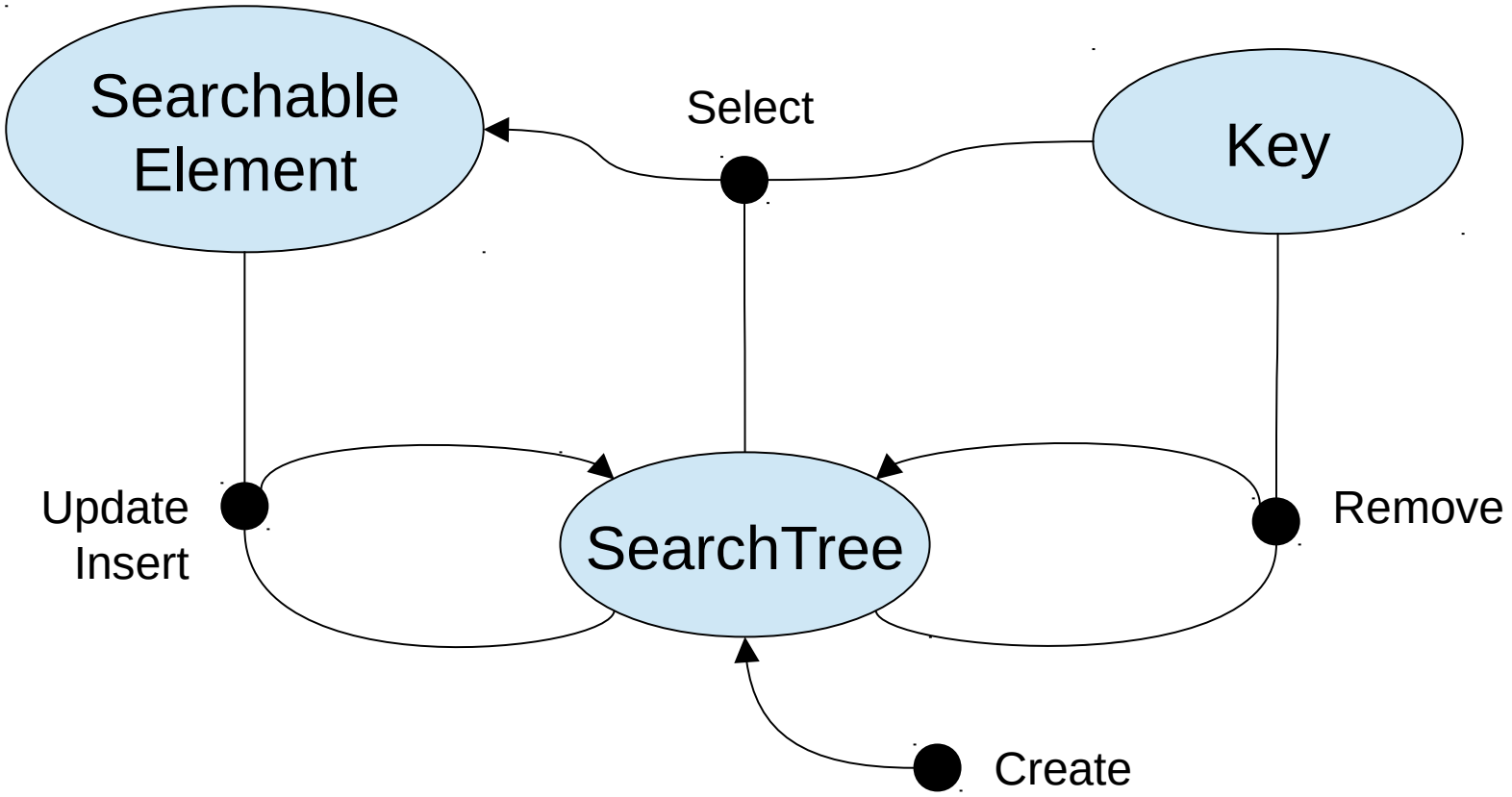


Příklad odebrání prvku z haldy

Stromy

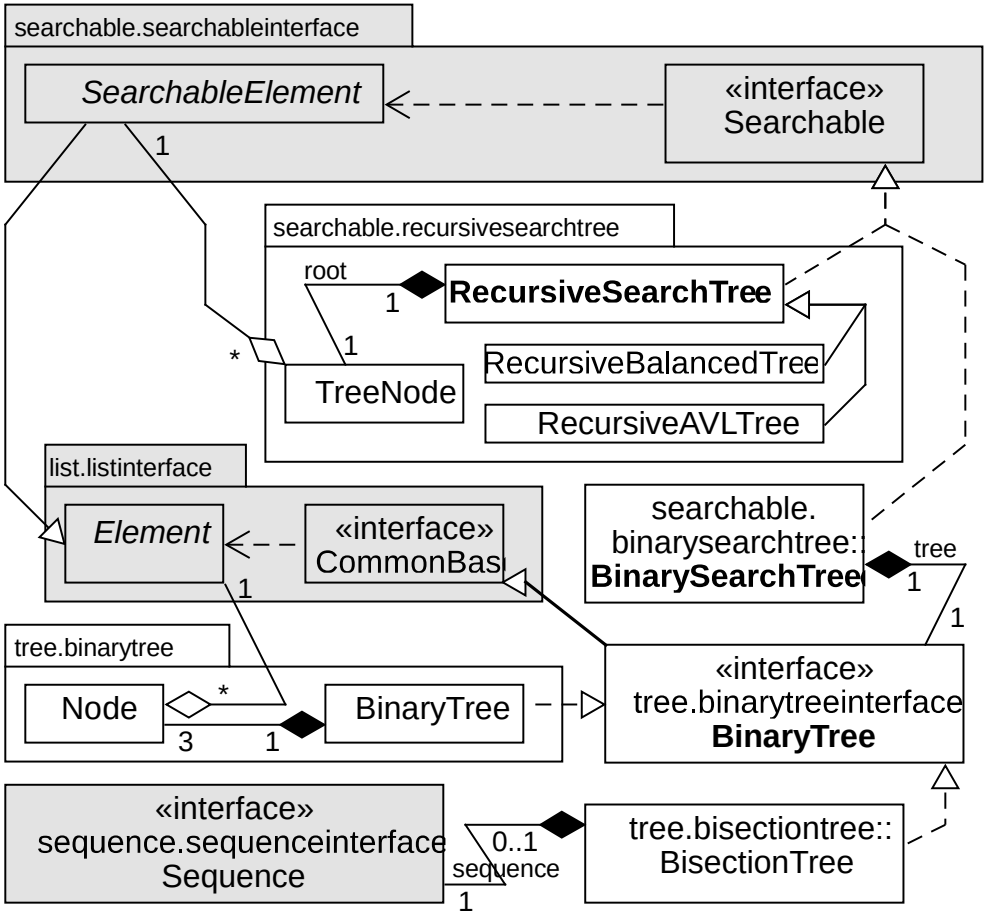
- Agenda
  - Pojmy graf a strom
  - Halda
  - **Binární vyhledávací strom**
  - Bisekce
  - Vyhledávání v binárním vyhledávacím stromě
  - Vyvážování binárních vyhledávacích stromů
  - Shrnutí

Binární vyhledávací strom



Signatura datového typu *vyhledávací strom*

Binární vyhledávací strom



Design různých implementací binárního vyhledávacího stromu

Stromy

- Agenda
  - Pojmy graf a strom
  - Halda
  - Binární vyhledávací strom
  - **Bisekce**
  - Vyhledávání v binárním vyhledávacím stromě
  - Vyvážování binárních vyhledávacích stromů
  - Shrnutí

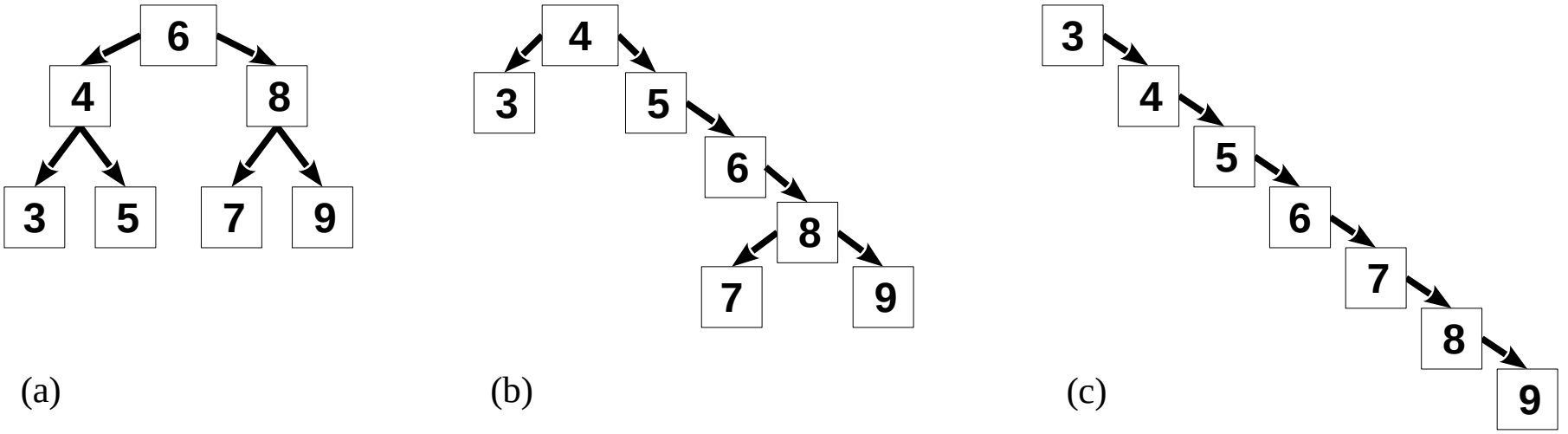
Bisekce

- Bisekce je způsob hledání v uspořádané posloupnosti.
  - Posloupnost nebudeme procházet od začátku prvek po prvku, nýbrž se v ní budeme pohybovat jako ve stromě:
    - prostřední prvek posloupnosti je kořen,
    - prostřední prvek spodní poloviny je kořen levého podstromu,
    - prostřední prvek horní poloviny je kořen pravého podstromu atd.
  - Nejprve porovnáme hledaný klíč s prostředním prvkem.
    - Je-li hledaný klíč menší, musí se hledaný prvek nacházet ve spodní polovině posloupnosti, tj. v levém podstromu.
    - Je-li naopak větší, tak analogicky v pravém podstromu.
  - Stejný postup potom zopakujeme ve zvolené polovině posloupnosti, pak ve čtvrtině, osmině atd. – až najdeme hledaný prvek nebo až zjistíme, že hledaný prvek v posloupnosti není.
- Abychom mohli tímto způsobem prohledávat posloupnost, potřebujeme objekt, který bude s posloupností zacházet jako se stromem.
  - Viz dokumentaci ke třídě `BisectionTree`
    - Implementuje rozhraní `BinaryTree`.
    - Je to jednoduchý obal (wrapper) nad posloupností.

Stromy

- Agenda
  - Pojmy graf a strom
  - Halda
  - Binární vyhledávací strom
  - Bisekce
  - **Vyhledávání v binárním vyhledávacím stromě**
  - Vyvážování binárních vyhledávacích stromů
  - Shrnutí

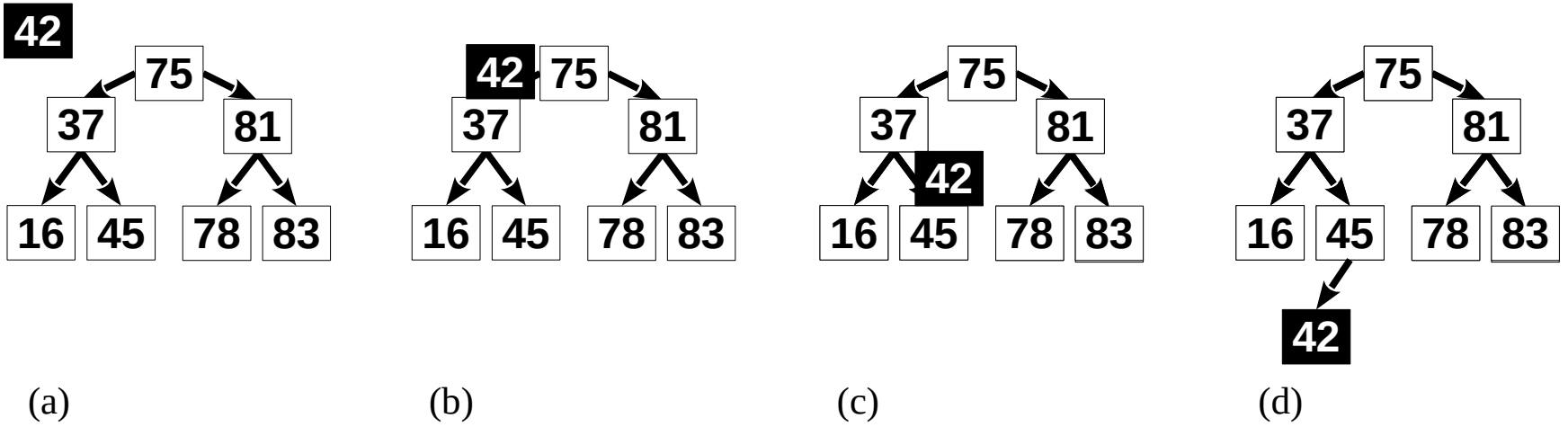
Vyhledávání



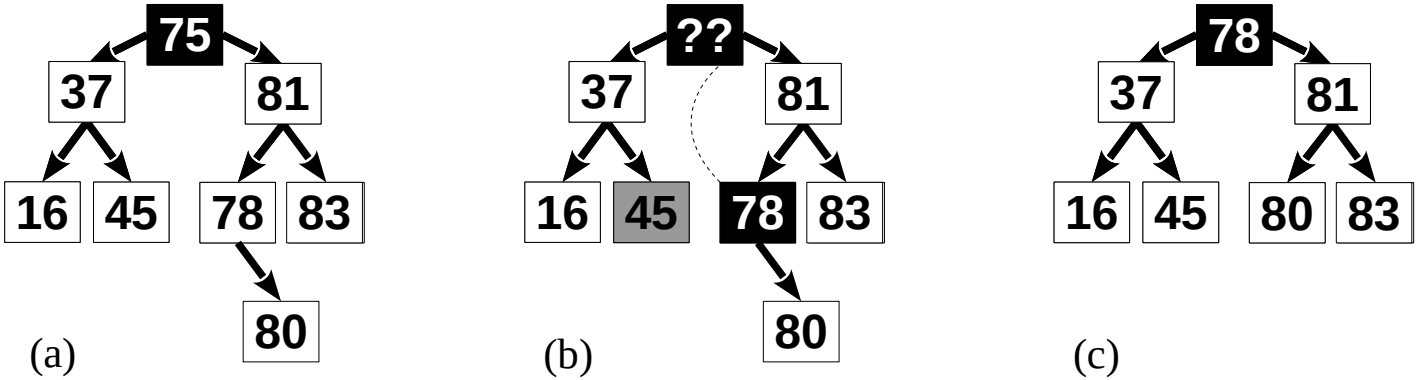
Příklady binárních vyhledávacích stromů

- Strom může degenerovat až na posloupnost.
- Pak je časová složitost operací lineární.

Vyhledávání



Příklad vložení prvku do binárního vyhledávacího stromu



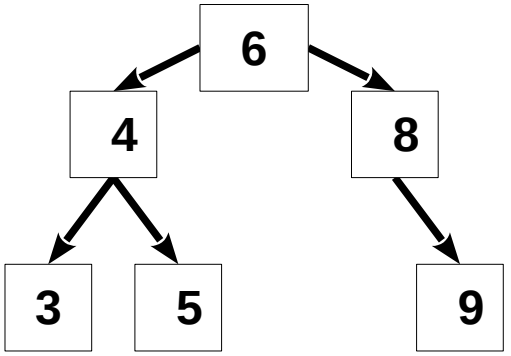
Příklad odstranění prvku z binárního vyhledávacího stromu

Stromy

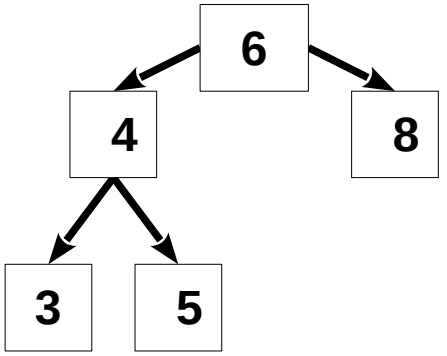
- Agenda
  - Pojmy graf a strom
  - Halda
  - Binární vyhledávací strom
  - Bisekce
  - Vyhledávání v binárním vyhledávacím stromě
  - **Vyvážování binárních vyhledávacích stromů**
  - Shrnutí

Vyvažování

- Při dokonalém vyvážení se počet vrcholů v levém a pravém podstromu smí lišit nejvýše o 1 vrchol.
- Při AVL-vyvážení se podstromy neliší o víc než o jednu vrstvu.



(a) dokonalé vyvážení



(b) AVL-vyvážení, které není dokonalé

Příklady vyvážení stromu

Vyvažování

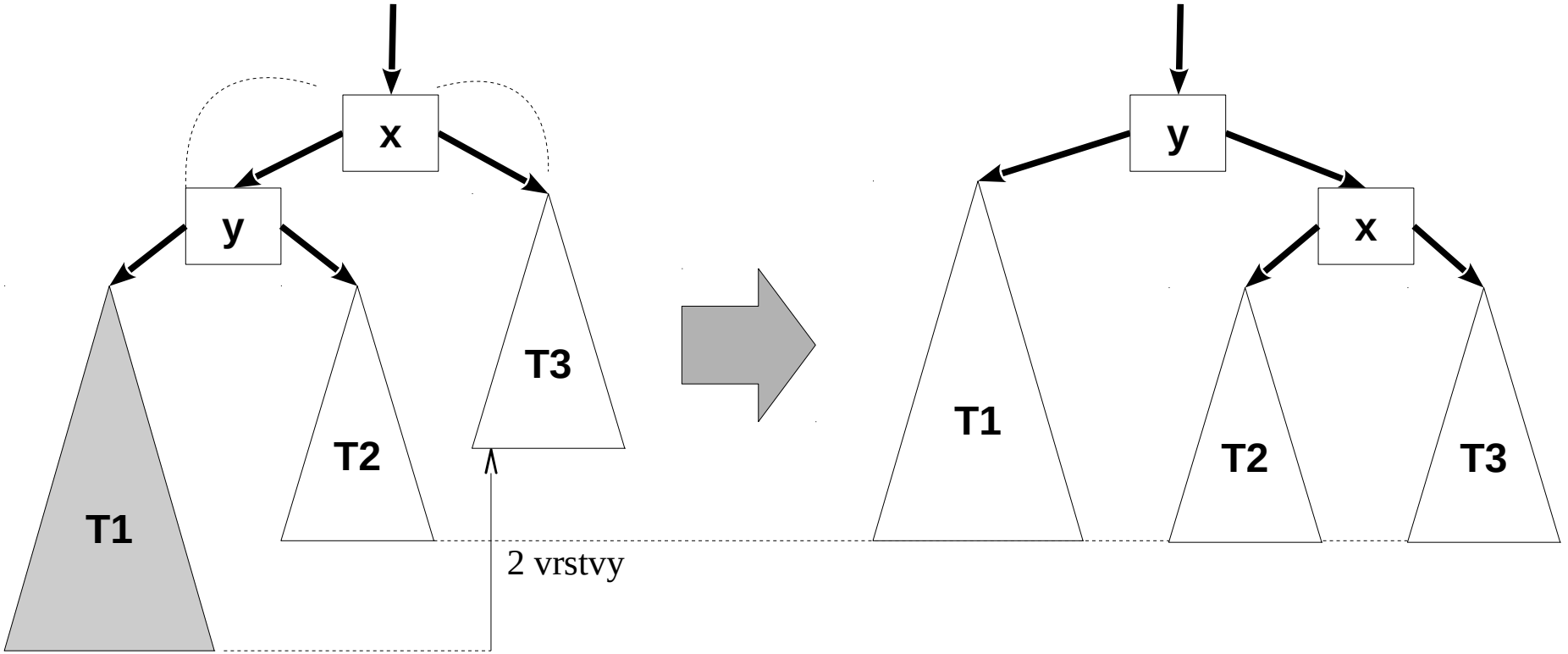


Schéma vyvážení AVL-stromu RR-rotací

Vyvažování

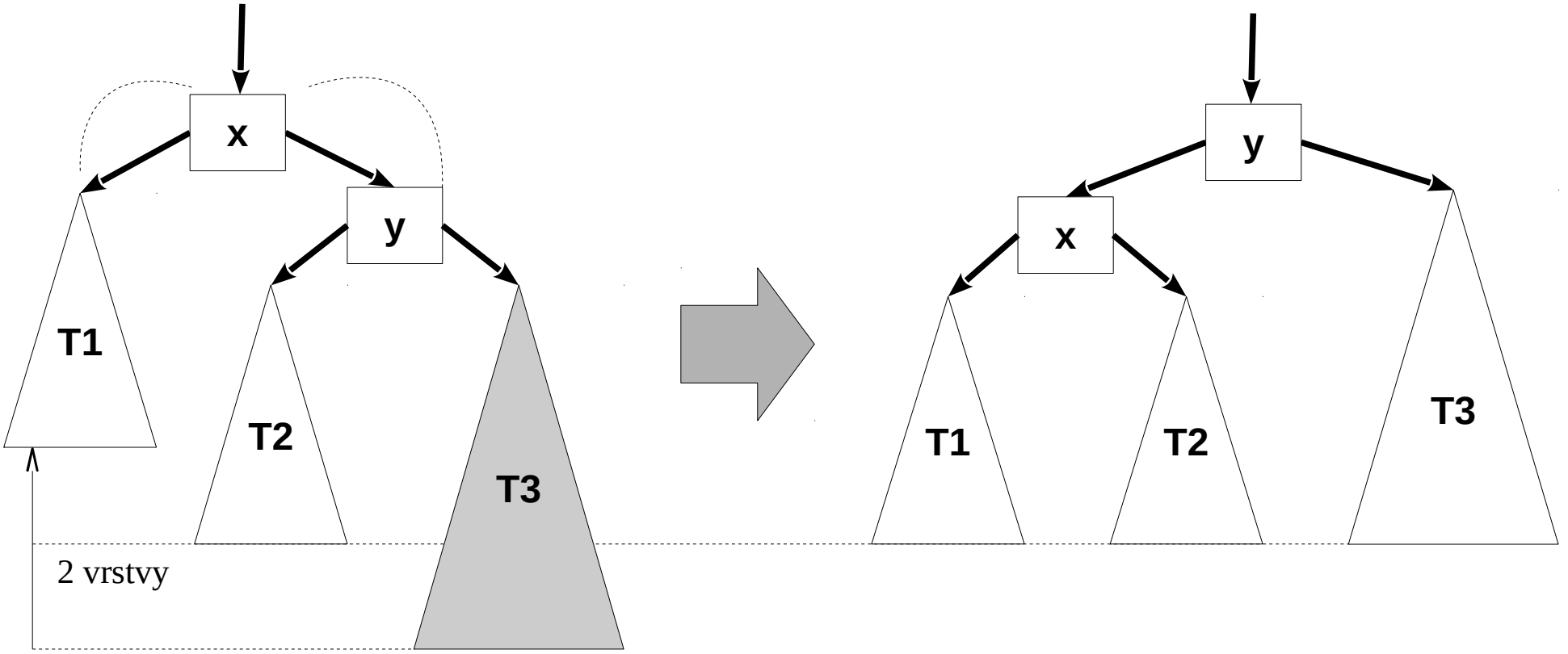


Schéma vyvážení AVL-stromu LL-rotací

Vyvažování

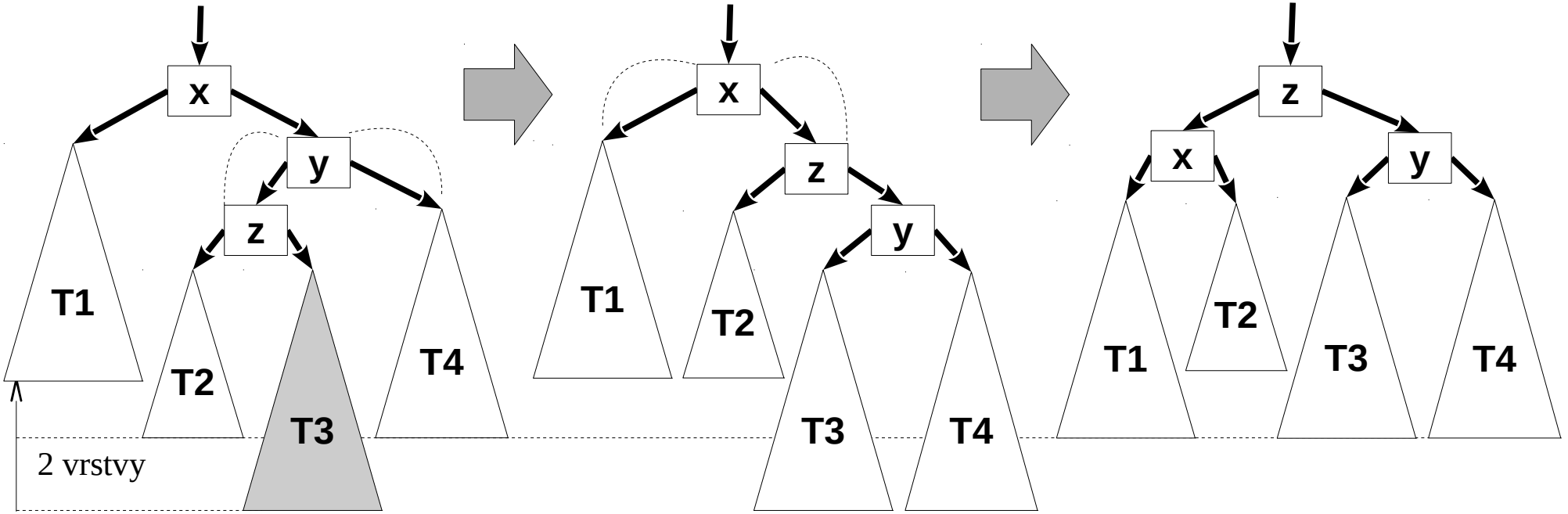


Schéma vyvážení AVL-stromu RL-rotací

Vyvažování

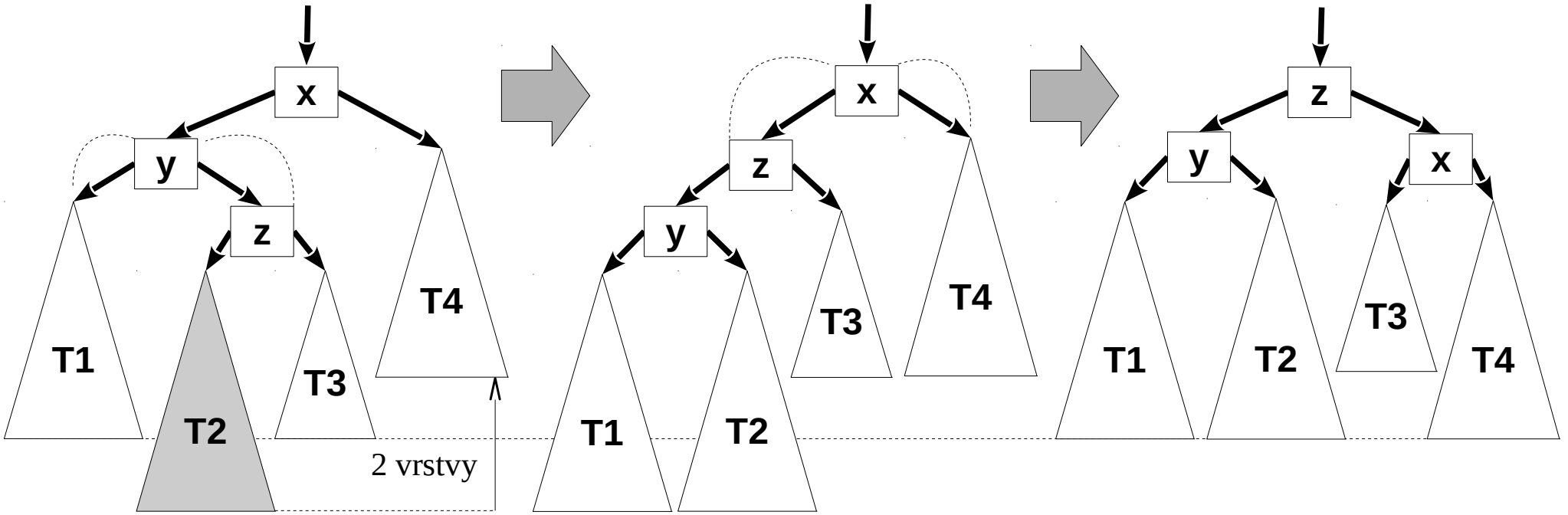


Schéma vyvážení AVL-stromu LR-rotací

Stromy

- Agenda
  - Pojmy graf a strom
  - Halda
  - Binární vyhledávací strom
  - Bisekce
  - Vyhledávání v binárním vyhledávacím stromě
  - Vyvážování binárních vyhledávacích stromů
  - **Shrnutí**

Shrnutí

- **Graf** může být *orientovaný,* nebo *neorientovaný.* V obou případech to je uspořádaná trojice tvořená množinou vrcholů, množinou hran, incidenční funkcí.
  - Incidenční funkce orientovaného grafu rozlišuje výchozí a cílové vrcholy hran.
  - Incidenční funkce neorientovaného grafu nerozlišuje mezi výchozím a cílovým vrcholem hrany.
- **Strom** je souvislý acyklický neorientovaný graf. Zvláštní druhy stromu jsou:
  - **Halda**
  - **Binární vyhledávací strom**

Shrnutí

- **Halda** je binární strom, který je vyvážený na počet vrstev.
  - Operace Get vybere z haldy vždy prvek s nejmenším klíčem, pokud je halda uspořádána podle těchto pravidel:
    - Předek má menší klíč než kterýkoli z jeho potomků.
    - Všechny vrstvy kromě poslední musí být plné, halda je tedy vyvážená.
    - Spodní vrstva se zaplňuje bez vynechávání zleva doprava.
  - Haldu lze uspořádat i opačně, aby operace Get vybírala prvek s největším klíčem.
  - Časová složitost operací haldy je logaritmická.
  - Halda se snadno implementuje polem.

Shrnutí

- **Bisekcí** rychle vyhledáváme v seřazených posloupnostech, časová složitost je logaritmická.
- **Binární vyhledávací strom** rychle vyhledává bisekcí a také rychle vkládá a odstraňuje prvky.
  - Prvky s hodnotou klíče menší než klíč vrcholu se umístí do levého podstromu tohoto vrcholu.
  - Prvky s hodnotou klíče větší než klíč vrcholu se umístí do pravého podstromu tohoto vrcholu.
  - Obyčejný vyhledávací strom může degenerovat až na posloupnost. Rychlost operací pak degraduje až na lineární časovou složitost.
  - Degeneraci se bráníme vyvažováním stromu
    - dokonale (podstromy každého vrcholu musí mít téměř stejný počet prvků)
    - AVL (podstromy každého vrcholu musí mít téměř stejný počet vrstev – to je chytřejší)
  - Vyvážené stromy zaručují, že časová složitost jejich operací nebude horší než logaritmická.

Konec

Tato prezentace patří k učebnici *Algoritmy a datové struktury objektově* od Ivana Ryanta. Obsahuje texty a obrázky z této učebnice.

RYANT, Ivan. *Algoritmy a datové struktury objektově.* Praha: Ivan Ryant, 2017. ISBN 978-80-270-1660-0