ADSO
9. Prohledávání do hloubky a do šířky

(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

Agenda

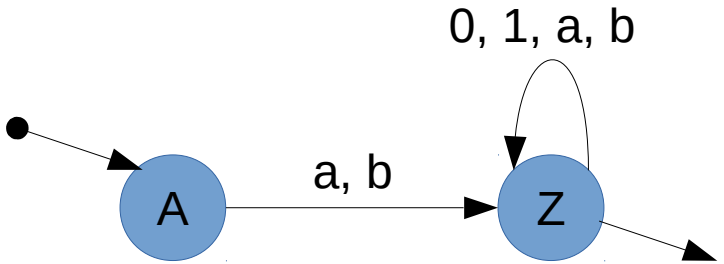
- **Prohledávání do hloubky a do šířky**
  - Stavový prostor
  - Prohledávání do hloubky
  - Prohledávání do šířky
  - Využití heuristiky
  - Shrnutí

Agenda

- Prohledávání do hloubky a do šířky
  - **Stavový prostor**
  - Prohledávání do hloubky
  - Prohledávání do šířky
  - Využití heuristiky
  - Shrnutí

Stavový prostor

- Stavový diagram konečného automatu
  - Názorný, ale počet stavů musí být konečný
- Obecně: **stavový prostor algoritmu** je jako konečné nebo nekonečné bludiště (~graf)
  - Stavy ~ křižovatky, rozcestí, vchody, východy, konce slepých chodeb (~vrcholy grafu)
  - Počáteční stav ~ vchod do bludiště
  - Cílové stavy ~ východy z bludiště, poklad (který hledáme) apod.

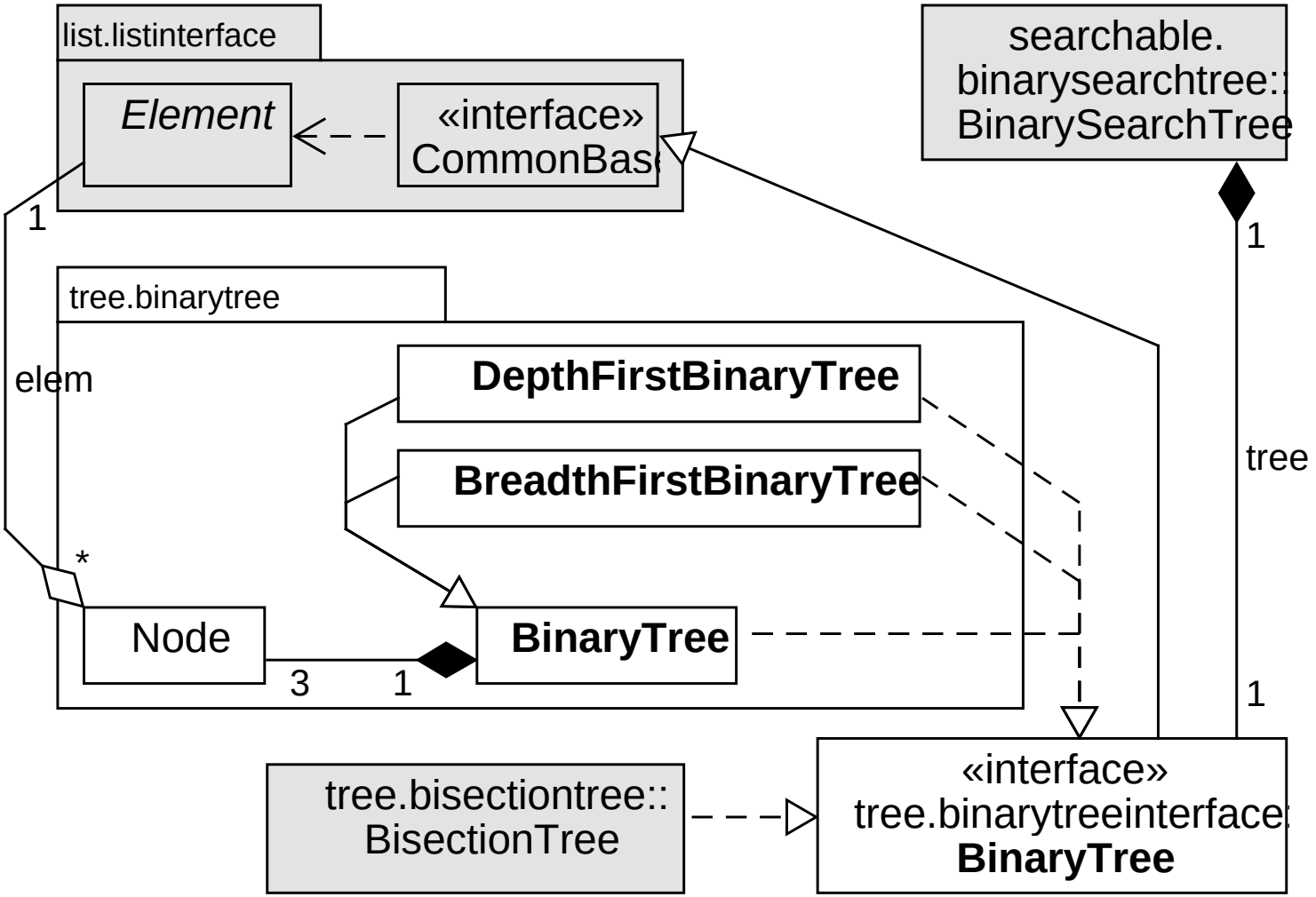


Zdroj:
https://en.wikipedia.org/wiki/Maze?oldid=723971883#/media/File:Longleat_maze.jpg

Stavový prostor

- Pozor: obecný graf může obsahovat cykly
  - musíme zabránit bloudění v kruhu!
  - Např. si označíme navštívené vrcholy
- Algoritmus prohledávání stavového prostoru je jeden z nejobecnějších algoritmů, umí řešit každou algoritmicky řešitelnou úlohu.
- **Pozor na velkou časovou náročnost!**

Stavový prostor



Design tříd procházejících strom do hloubky a do šířky

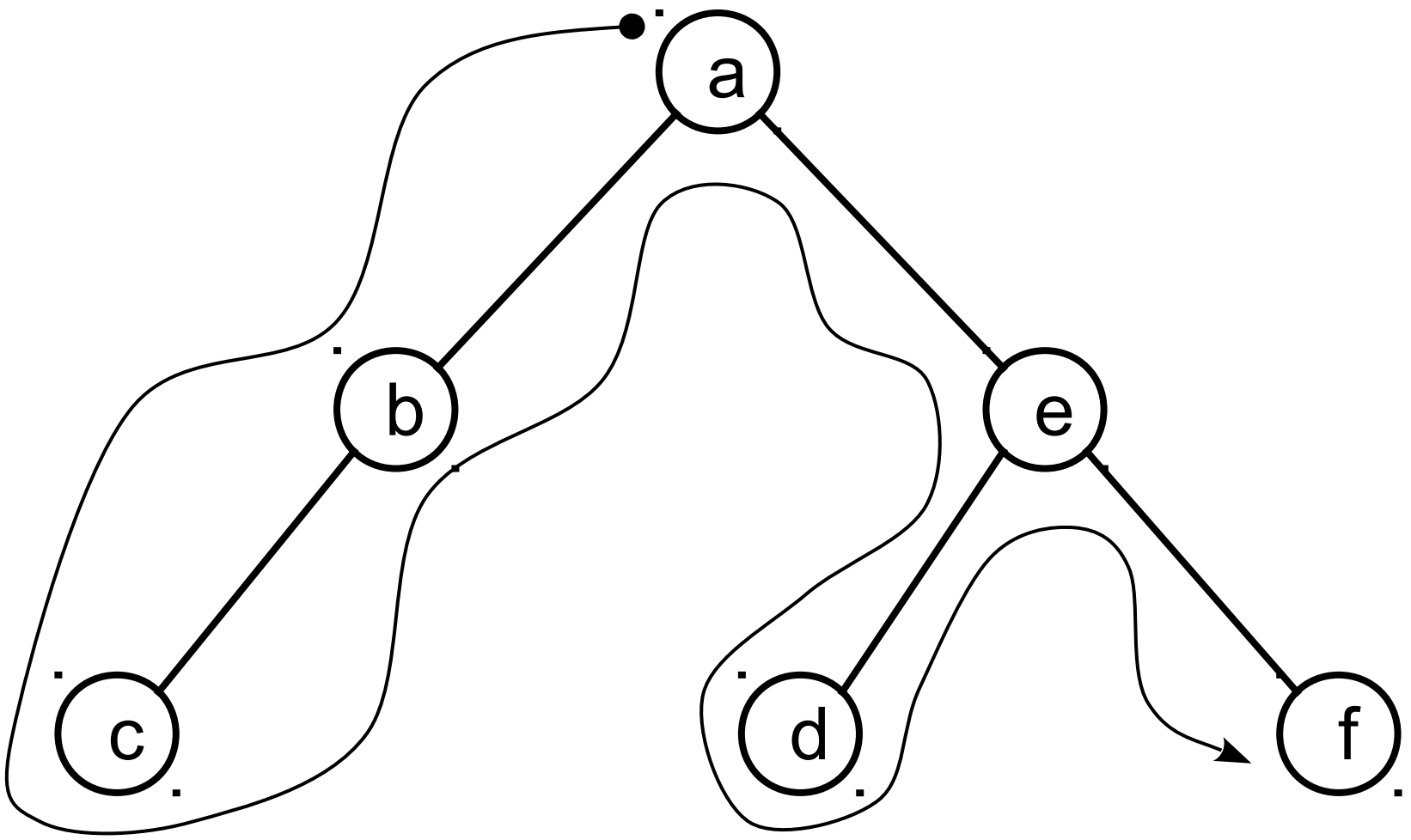
Agenda

- Prohledávání do hloubky a do šířky
  - Stavový prostor
  - **Prohledávání do hloubky**
  - Prohledávání do šířky
  - Využití heuristiky
  - Shrnutí

Prohledávání do hloubky

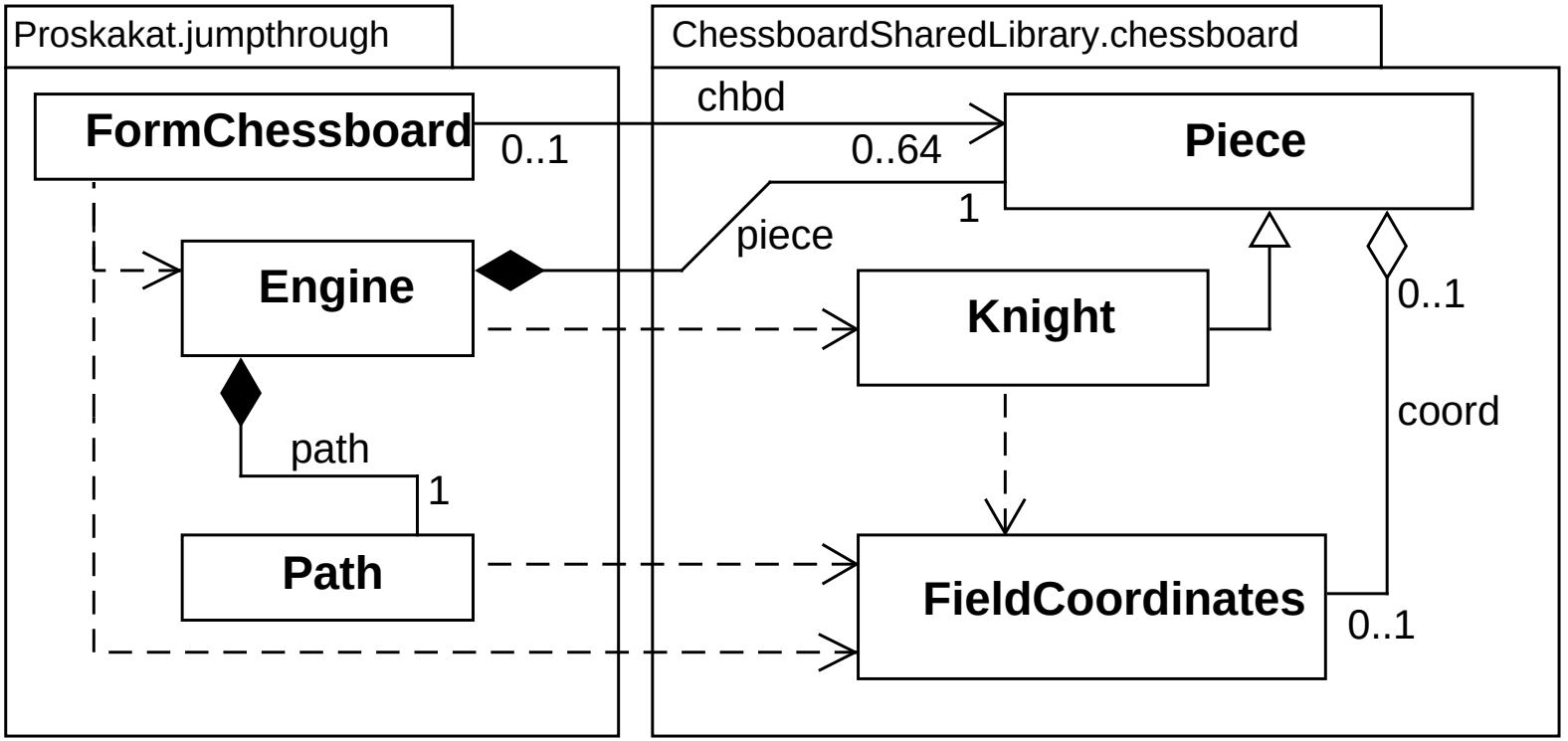
- Prohledávání binárního stromu začíná v kořenu.
- V kořenu i v každém dalším navštíveném vrcholu
  - nejprve prohledáme levý podstrom
  - pak vykonáme nějakou užitečnou činnost s vrcholem
  - nakonec prohledáme pravý podstrom.
- Levý i pravý podstrom prohledáme stejným způsobem
  - z toho plyne, že nejjednodušší implementace tohoto algoritmu bude využívat rekurzi.
- Analogicky můžeme prohledávat i jiné než *binární* stromy, příp. *grafy* obecně (pozor na cykly!)
- Tento rekurzivní algoritmus **prohledávání s návraty** se obvykle nazývá anglickým termínem **backtracking.**

Prohledávání do hloubky



Schematické znázornění, jak se prohledává do hloubky

Prohledávání do hloubky



Design systému, který simuluje průchod koně šachovnicí

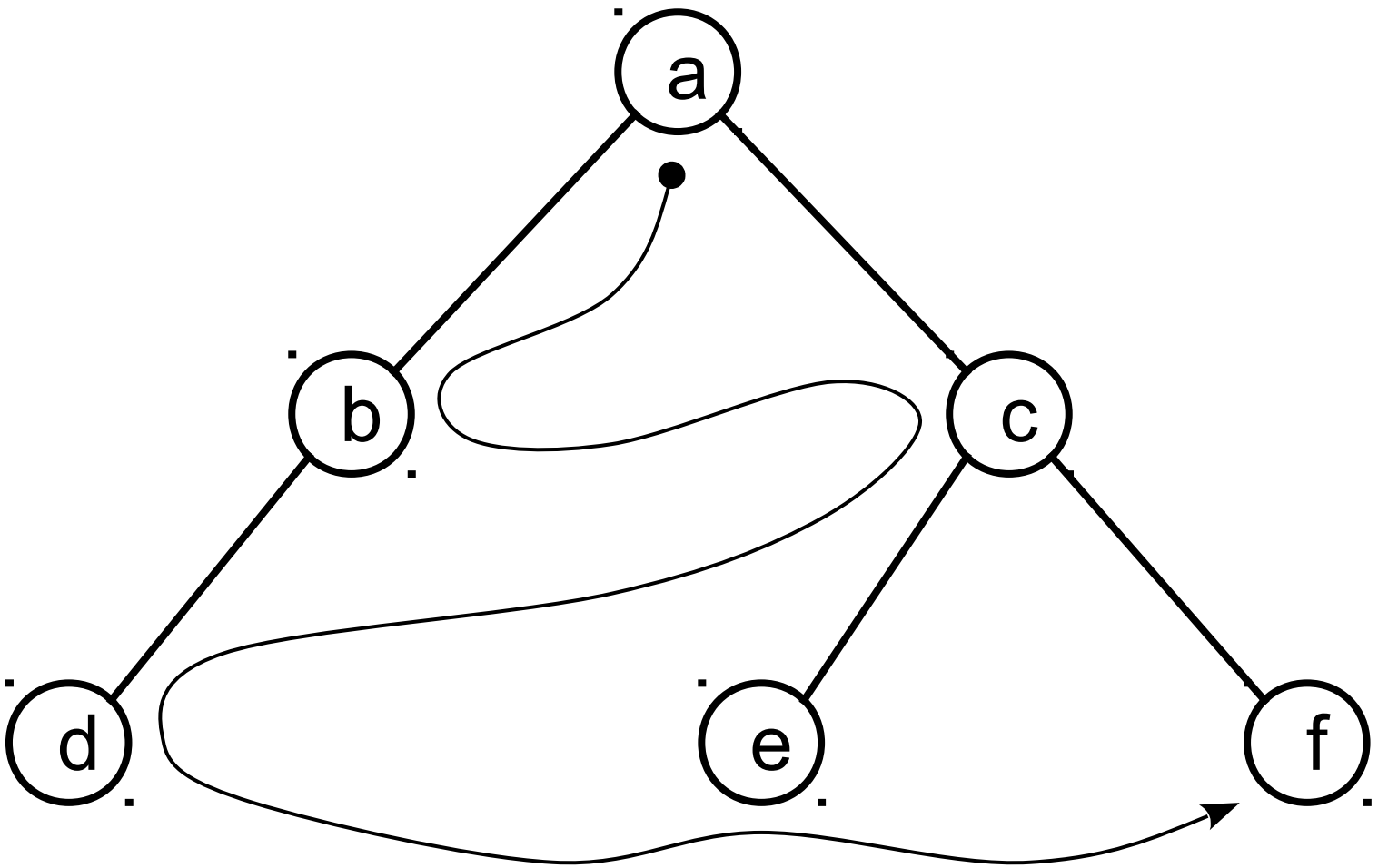
Agenda

- Prohledávání do hloubky a do šířky
  - Stavový prostor
  - Prohledávání do hloubky
  - **Prohledávání do šířky**
  - Využití heuristiky
  - Shrnutí

Prohledávání stromu do šířky

- Procházíme napřed kořen, pak druhou vrstvu vrcholů, a to zleva doprava.
- Tak pokračujeme vrstvu po vrstvě, až projdeme všechny vrstvy a tím i všechny vrcholy.
- Jak zajistíme procházení vrcholů ve správném pořadí?
  - Pomůžeme si **frontou.**
    - Napřed vložíme do fronty kořen stromu.
    - Jakmile máme ve frontě aspoň jeden vrchol, začneme odebírat vrchol po vrcholu a potomky každého odebraného vrcholu zařadíme na konec fronty.
    - Jakmile se fronta vyprázdní, prošli jsme celý strom.

Prohledávání do šířky



Schematické znázornění, jak se prohledává do šířky

Agenda

- Prohledávání do hloubky a do šířky
  - Stavový prostor
  - Prohledávání do hloubky
  - Prohledávání do šířky
  - **Využití heuristiky**
  - Shrnutí

Využití heuristiky

- Velkou časovou náročnost snížíme
  - Ořezáváním
    - Např. zkrácené vyhodnocování výrazů
      - Násobení nulou
      - Konjunkce s nepravdou
      - Apod.
  - Heuristikou
    - Např. proskákání šachovnice koněm:
      - Kůň napřed navštíví ta políčka, z nichž vede dál méně cest.
      - Tím se zpočátku zrychlí prohledávání cest, cesty časově náročnější se odloží na později.
      - Jestli hledaná cesta není jedna z posled-ních, najde se rychleji než bez heuristiky.

Agenda

- Prohledávání do hloubky a do šířky
  - Stavový prostor
  - Prohledávání do hloubky
  - Prohledávání do šířky
  - Využití heuristiky
  - **Shrnutí**

Shrnutí

- **Stavový prostor úlohy** je zvláštním případem **grafu.**
  - Je-li počet stavů konečný, dá se stavový prostor znázornit stavovým diagramem.
  - Prohledávání stavového prostoru je univerzální (ale neefektivní) způsob, jak můžeme najít řešení jakékoli algoritmicky řešitelné úlohy.
- Grafy prohledáváme jako by to byly stromy.
  - Algoritmy prohledávání můžeme použít i na *grafy s cykly.* Cyklus v grafu způsobí, že se algoritmus může dostat do již dříve navštíveného vrcholu podruhé. Když to zjistí, přestane znovu prohledávat graf za tímto vrcholem, aby se nezacyklil. Proto si musí pamatovat, které vrcholy již prohledal (např. si je označí jako navštívené).
- Jsou dva standardní postupy prohledávání:
  - **Do hloubky** (depth-first)
    - Prohledáváme s návraty (backtrack). Backtracking je rekurzivní algoritmus. Rekurzi můžeme nahradit cyklem a k uchovávání stavu prohledávání použít zásobník.
  - **Do šířky** (breadth-first)
    - Prohledáváme vlnou. Algoritmus obsahuje cyklus a k uchovávání čela vlny používá frontu.
- Prohledávání stavového prostoru bývá náročné časově i paměťově. Můžeme je urychlit např. *ořezáváním* (např. při zkráceném vyhodnocování logických výrazů) nebo různými *heuristikami* (viz příklad o proskákání šachovnice).

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