

ADSO

8. Rozptýlené tabulky

(prezentace k učebnici)

Ivan Ryant

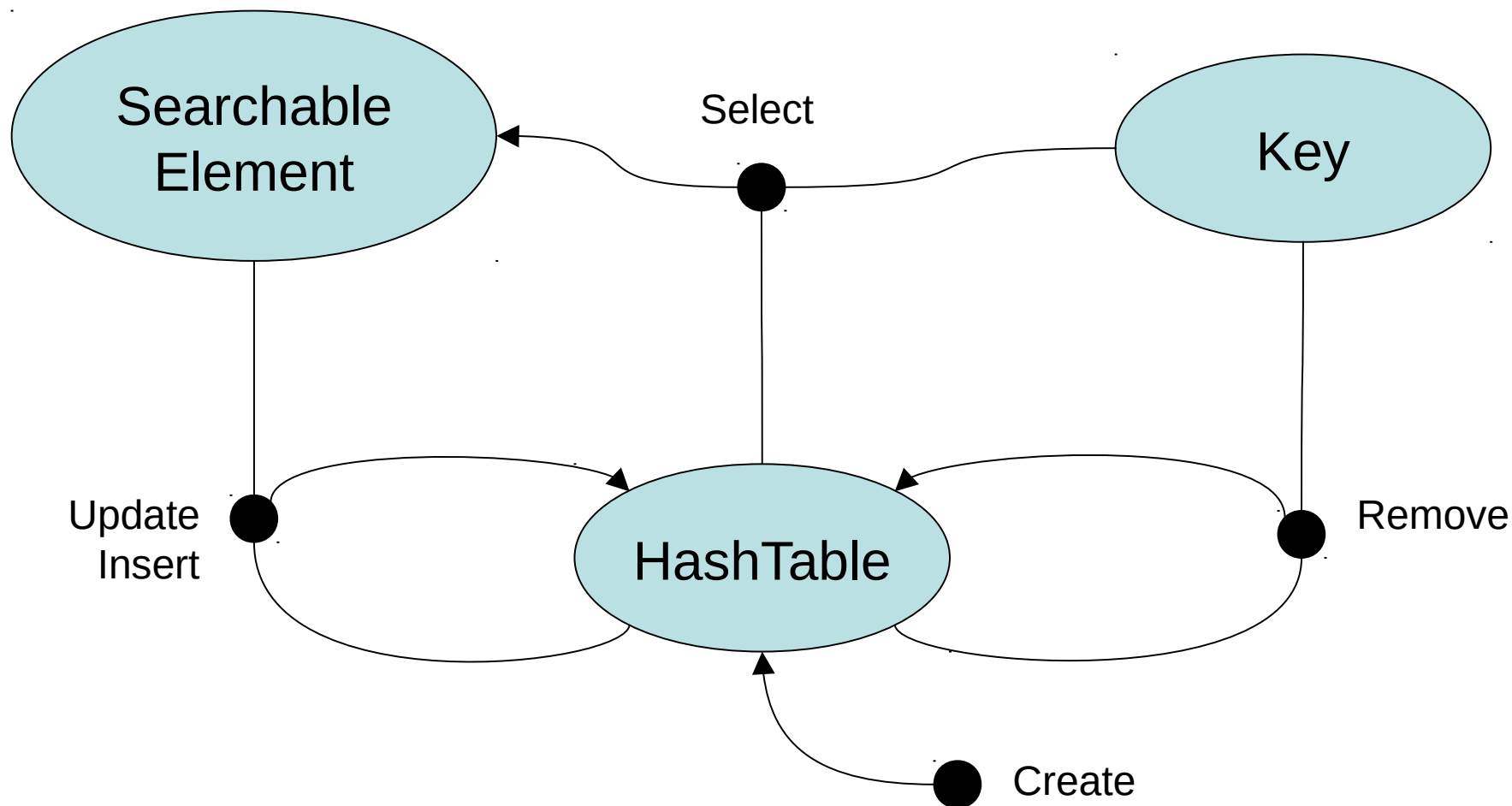
Agenda

- 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

- **Rozptýlené tabulky**
 - princip rozptýlené tabulky, hašování
 - tabulka s vnějším zřetězením
 - tabulka s vnitřním zřetězením

Rozptýlené tabulky

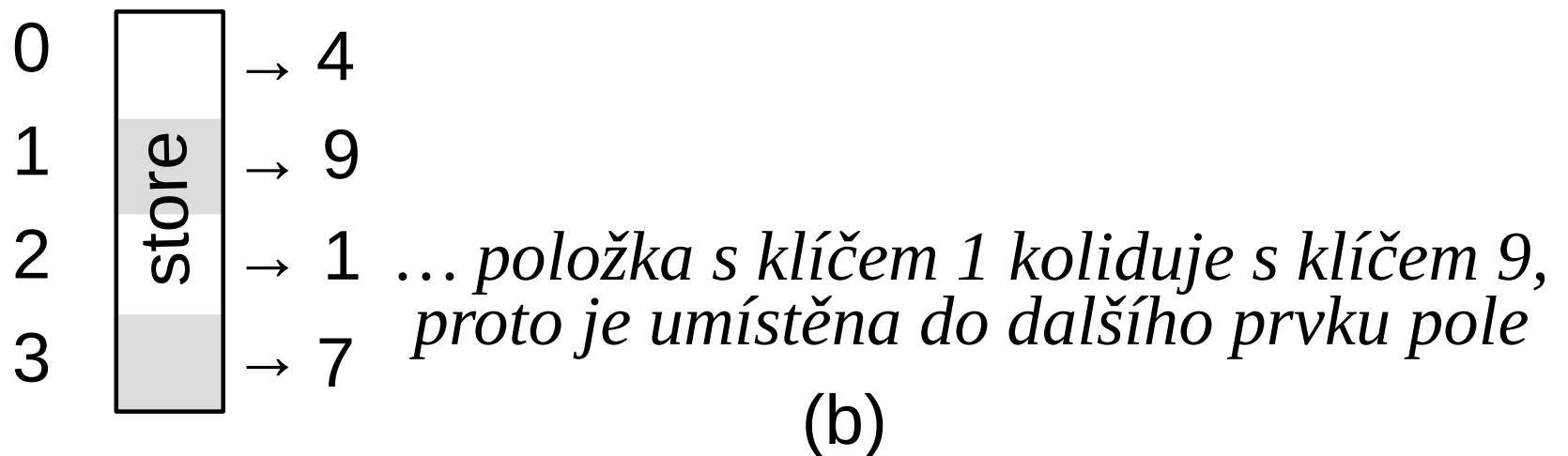
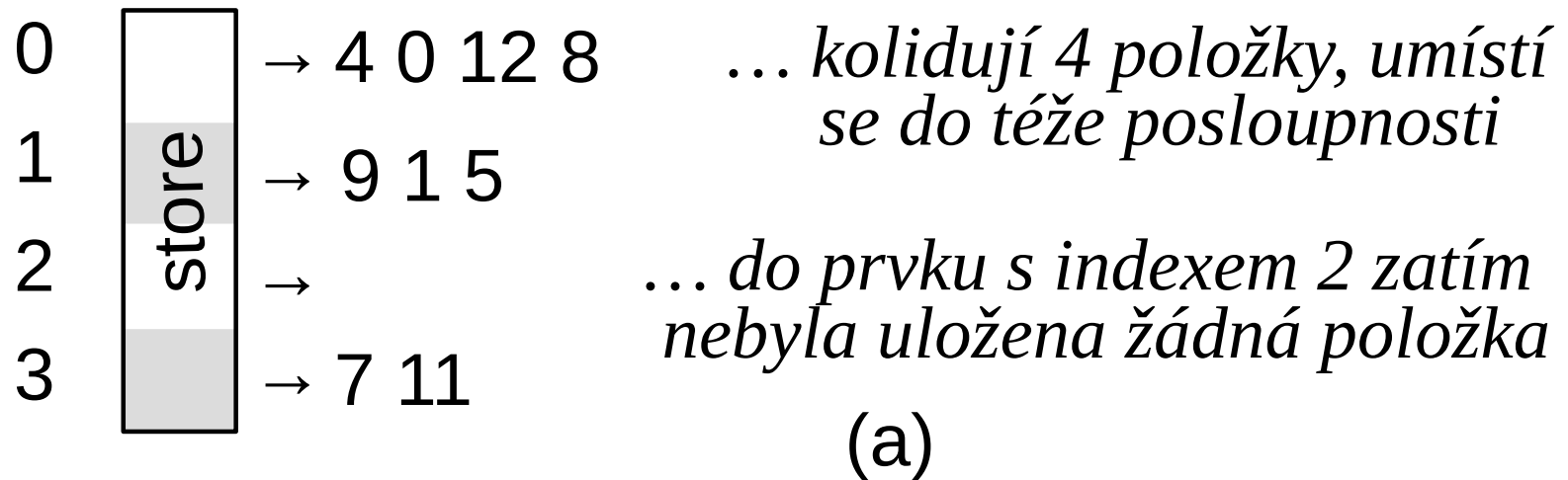


Signatura datového typu *Rozptýlená tabulka*

Rozptýlené tabulky

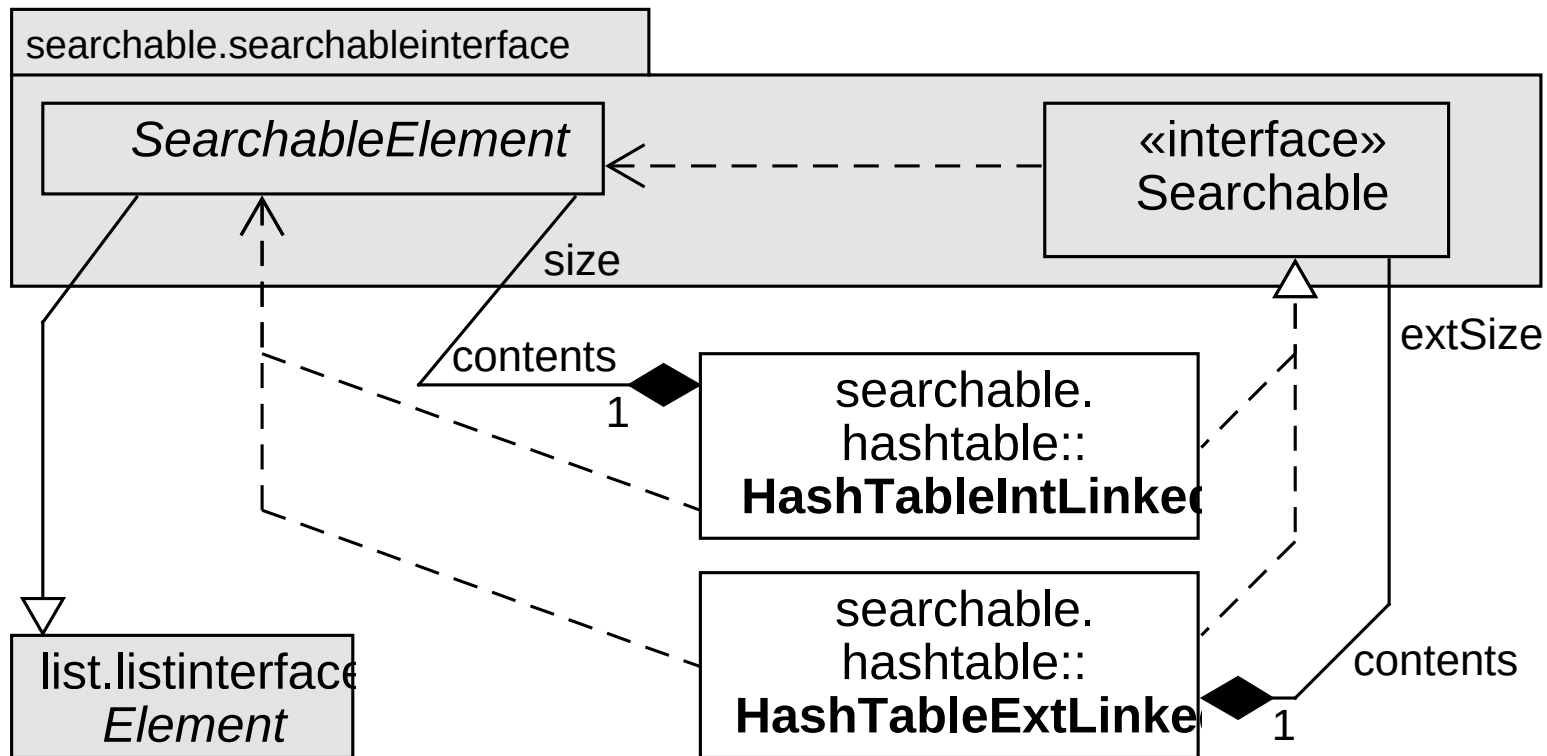
- Položky tabulky typicky ukládáme do pole – dají se indexovat (rychlost).
- Neindexujeme přímo klíči (kvůli velikosti pole). Klíče prvků přepočítáme na indexy **transformační (hašovací) funkcí**. Tuto funkci volíme:
 - univerzálně, ale bez ohledu na efektivitu (např. jako zbytek po dělení)
 - heuristicky (celkově efektivní, ale v jednotlivých případech bez záruky).
- Transformační (hašovací) funkce:
 - mapuje klíče do rozsahu indexů
 - může mapovat různé klíče na týž index – **kolize** se řeší:
 - Prvek pole nemusí obsahovat jen jednu položku, ale celou vyhledávací strukturu, do které umístíme všechny kolidující položky – **vnější zřetězení**.
 - Nová položka se neumístí do obsazeného prvku, nýbrž do některého volného prvku pole – **vnitřní zřetězení**.

Rozptýlené tabulky



Příklad vnějšího a vnitřního zřetězení rozptýlené tabulky

Rozptýlené tabulky



Design různých implementací rozptýlené tabulky

Shrnutí

- Rozptýlená tabulka (hashtable) je dalším typem vyhledávací struktury. Její vnější chování se od ostatních vyhledávacích struktur obvykle liší jen daleko rychlejšími operacemi.
- Při hledání položky se klíč přepočítá **transformační (hašovací) funkcí** na index pole.
- Jestliže se různé klíče několika prvků mapují na týž index, nastává **kolize**. Kolize se řeší dvojím způsobem:
 - **Vnější zřetězení:** každý prvek pole obsahuje další vyhledávací strukturu, např. vyhledávací strom nebo posloupnost.
 - **Vnitřní zřetězení:** kolidující položky budou umístěny do jiných, dosud neobsazených prvků pole, než určil vypočtený index.
 - Časová složitost vyhledávání může být v nejhorším případě až lineární vzhledem k počtu prvků.
 - Kolize při vnitřním zřetězení se řeší *heuristicky* s cílem, aby v průměru byla složitost vyhledávání téměř konstantní (není-li tabulka příliš zaplněná).
 - Transformační neboli hašovací funkce je také *heuristická*. Cílem je, aby ke kolizím docházelo jen výjimečně a vyhledávání bylo většinou rychlé.
- Rozptýleným tabulkám konkurují co do rychlosti *reorganizované posloupnosti*.

Konec

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

Tato prezentace smí být volně šířena, ale vždy i s tímto snímkem. Citujete-li, vyznačte zřetelně citát v plném rozsahu a uveďte zdroj:

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