

## Tabulka

Datová struktura, která umožňuje vkládat a později vybírat informace podle identifikačního klíče.

- Pevně definované (LUT Look Up Table)
- S proměnným počtem položek

Konvence:

- $k$  – klíč
- $A_k$  – adresní klíč (index)

### Tabulky s přímým přístupem

$k \rightarrow A_k$  je prosté zobrazení, každá položka tabulky má své místo jednoznačně určené

optimální implementace tabulky je pomocí pole – indexy jsou přímo klíče v tabulce

Telefonní seznam – klíčem je jméno

### Vyhledávací tabulky

Vyhledává se podle hodnoty klíče

Pořadí položek může být

- Definované (uspořádané)
- Náhodné

Strategie vyhledávání:

- Sekvenční
- Binární
- Fibonacciho
- Kombinované

## Tabulky s rozptýlenými položkami

Používají se v případě, že rozsah kláče  $N \gg$  rozsah tabulky

Pro určení pozice v tabulce, na kterou máme uložit položku s klíčem  $k$ , používáme rozptylovací funkci (hash-funkci), která jednoznačně přiřazuje klíč

### Tabulka s otevřeným rozptýlením

Každá pozice tabulky je potenciálně přístupná položce s libovolným klíčem, snadno vznikají shluky položek. Při kolizi se hodnota indexu přepočítá. Rozptylovací funkci řešíme například modulo  $p$  ( $p$  je prvočíslo)

Podle způsobu řešení kolizí rozeznáváme čtyři podtypy:

1. Tabulky s otevřeným rozptýlením a *nedefinovaným způsobem* ukládání synonymických položek

– nutno dodefinovat způsob přepočítání  $A_k$  při kolizi

*Příklad:*

32	0		
	1		
	2	???	
38	3		
	4		
	5		

Kolize: prvky 32 a 38 mají stejnou hodnotu rozptylové funkce

2. Tabulky s otevřeným rozptýlením a ukládáním synonymických položek s *konstantním krokem*  $s$  ( $s$  vícenásobnou hashovací funkcí)

– na klíč  $k$  aplikujeme funkci  $h_0(k)$ , pokud dojde ke kolizi, vypočteme znovu  $A_k$  podle  $h_1(k)$  atd. až do té doby, než najdeme v tabulce volné místo

$$\begin{aligned}
 h_0(k) &= h(k) \\
 h_1(k) &= (h(k) + s) \bmod p \\
 &\vdots \\
 h_i(k) &= (h(k) + is) \bmod p
 \end{aligned}$$

kde  $p$  je rozsah tabulky a  $s$  je přirozené číslo nesoudělné s  $p$  (nesoudělnost zaručí možnost dostat se na každou pozici tabulky)

*Příklad:* Tabulka s

- 8 pozicemi
- položkami s klíči DAVID, HANA, DANA, HELENA, EMIL, EVA, BOŽENA
- hashovací funkcí

$$h(k) = \text{pořadí 1. písmene v abecedě}$$

S = 1			S = 3		
0	HELENA	2	0		
1	BOŽENA	1	1	BOŽENA	1
2			2	HELENA	2
3	DAVID	1	3	DAVID	1
4	DANA	2	4	EMIL	1
5	EMIL	2	5	EVA	4
6	EVA	3	6	DANA	2
7	HANA	1	7	HANA	1
st. složitost = 12/7			st. složitost = 12/7		

3. Tabulky s otevřeným rozptýlením a ukládáním synonym s *lineární vícenásobnou ukládací funkcí*

$$\begin{aligned} h_0(k) &= h(k) \\ h_i(k) &= (h(k) + a \cdot i + b) \bmod p \end{aligned}$$

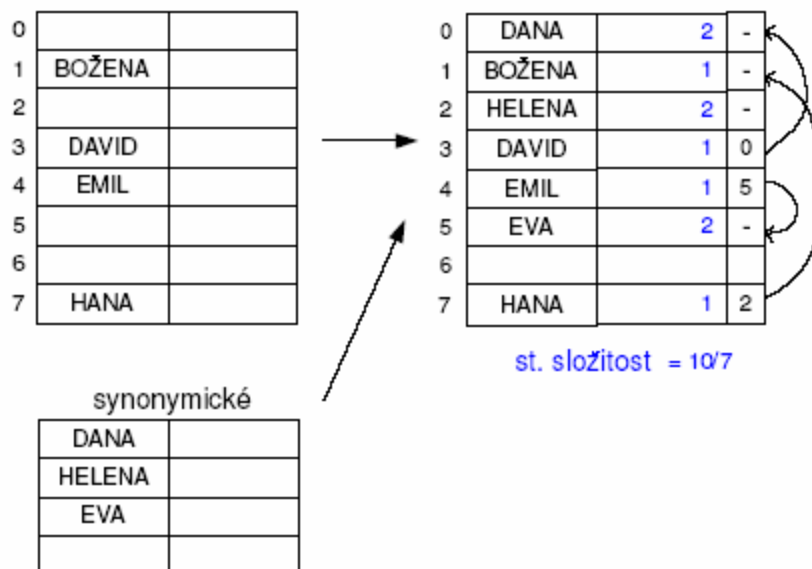
4. Tabulky s otevřeným rozptýlením a ukládáním synonym s *kvadratickou vícenásobnou ukládací funkcí*

$$\begin{aligned} h_0(k) &= h(k) \\ h_i(k) &= \left( h(k) + (-1)^i \cdot \left\lceil \frac{i}{2} \right\rceil^2 \right) \bmod p \end{aligned}$$

kde  $\lceil x \rceil$  značí zaokrouhlení čísla  $x$  „nahoru“. U zjišťování zbytku po dělení (*mod*) je třeba dávat pozor na záporné argumenty — držet se definice!

## Tabulky s otevřeným rozptýlením a vnitřním zřetězením

Ve fázi zařazování nových položek do tabulky si při výskytu kolize ponecháme synonymické položky vedle. Po zařazení všech položek vřadíme tato dočasně odložená synonyma na zbylá volná místa v tabulce (podle předem zvoleného systému) a příslušná synonyma zřetězíme do tzv. *řetězu synonym*



Přesun synonym do tabulky:

1. od začátku
  2. od konce
  3. s vícenásobnou hash-funkcí (to je výhodné, protože nám to „nerozbourá“ rovnoměrné rozložení prvků)
- tento typ tabulky je vhodný, pokud tabulku na začátku jednou vytvoříme a potom opakovaně používáme
  - problém je přidání položky na první místo v řetězci synonym, pokud toto místo již jiná položka obsadila.

## Tabulky s uzavřeným rozptýlením a vnějším zřetězením

– přinášejí zlepšení poslední metody: rozdělíme tabulku na dvě části

1. *primární část* — obsahuje pouze položky, které nejsou synonyma
2. *sekundární část* (*zóna zřetězení, přeplnění, atd.*) — položky, které jsou synonymy k položkám v primární části

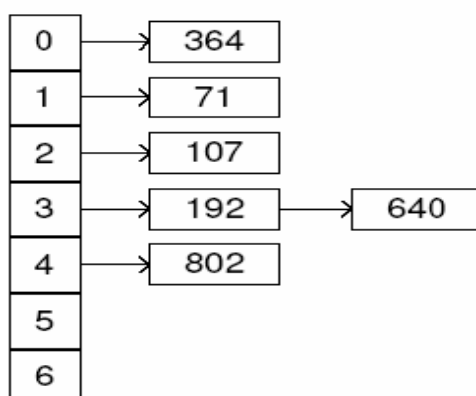
	0			-
	1	BOŽENA		
	2			
prim. část	3	DAVID		8
	4	EMIL		10
	5			
	6			
	7	HANA		9
	8	DANA		11
sek. část	9	HELENA		-
	10	EVA		-
	11	DIANA		-

zóna zřetězení

– problémem je vypuštění prvního prvku v řetězci — řeší se zavedením *děř*

### Metoda rozptýlených indexů

– tabulku chápeme jako vektor seznamů synonymických položek (řetězců)



– zařazování nových prvků:

1. Na konec řetězu
2. Uspořádaně – složitější implementace, ale přináší jisté urychlení

-velice oblíbená metoda (100% plnění, jednoduchá implementace, ...)