

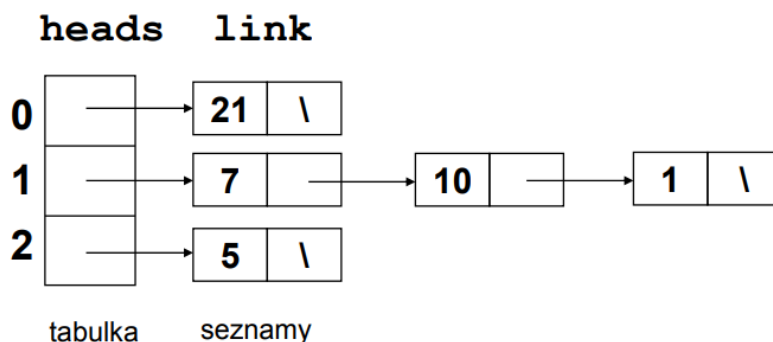
25. Zřetězené hashování, otevřené hashování, linear probing, double hashing - principy, výhody/nevýhody atd.

Zřetězené hashování

- Adresy v hashovací tabulce obsahují lineární seznamy
- V případě kolize (stejná adresa) se prvek vloží na konec seznamu
- V případě hledání sekvenčně procházíme konkrétní seznam

Zřetězené hashování - příklad

- $h(k) = k \bmod 3$
- posloupnost : 1, 5, 21, 10, 7



Otevřené hashování

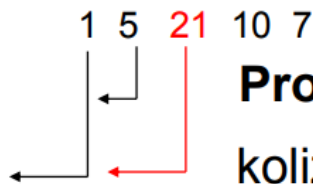
- Tabulka adres uložená do pole
- V případě kolize prohledáváme určitou metodou další prvky pole, dokud nenajdeme prázdnou pozici
- Při vyhledávání postupujeme stejně – stejnou metodou procházíme, pokud najdeme volnou pozici, znamená to, že prvek není indexován
- Podle metody hledání volného místa rozlišujeme:
 - Lineární prohledávání (linear probing)
 - Dvojí hašování (double hashing)

Otevřené hašování

$$h(k) = k \bmod 5$$

posloupnost:

0	5
1	1
2	
3	
4	



$$(h(k) = k \bmod m, m \text{ je rozměr pole})$$

Problém:

kolize - 1 blokuje místo pro 21
nasadíme:

1. linear probing
2. double hashing

Hledání volné pozice – probing

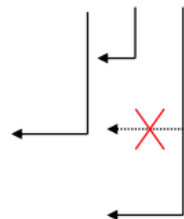
- Určení, zda pozice v tabulce obsahuje klíč shodný s hledaným klíčem
- Možnosti:
 - Search hit = klíč nalezen
 - Search miss = pozice prázdná, klíč nenalezen
 - Jinak = na pozici je jiný klíč, hledej dál

Linear probing

$$h(k) = [(k \bmod 5) + i] \bmod 5 = (k + i) \bmod 5$$

posloupnost: 1 5 21 10 7

0	5
1	1
2	21
3	
4	



kolize - 1 blokuje

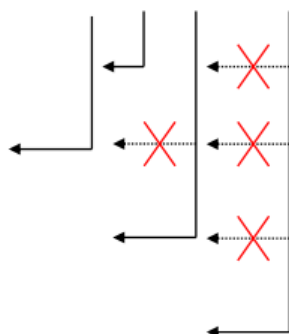
=> 1. linear probing

vlož o 1 pozici dál ($i++ \Rightarrow i = 1$)

$$h(k) = (k + i) \bmod 5$$

posloupnost: 1 5 21 10 7

0	5
1	1
2	21
3	10
4	

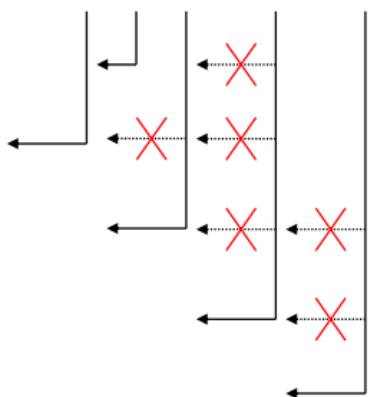


1. kolize - 5 blokuje - vlož dál
 2. kolize - 1 blokuje - vlož dál
 3. kolize - 21 blokuje - vlož dál
- vloženo o 3 pozice dál ($i = 3$)

$$h(k) = (k + i) \bmod 5$$

posloupnost: 1 5 21 10 7

0	5
1	1
2	21
3	10
4	7



1. kolize - vlož dál ($i++$)
 2. kolize - vlož dál ($i++$)
- vlož o 2 pozice dál ($i = 2$)

$$h(k) = (k + i) \bmod 5$$

posloupnost: 1 5 21 10 7

0	5	$i = 0$
1	1	$i = 0$
2	21	$i = 1$
3	10	$i = 3$
4	7	$i = 2$

Dvojíte hešování

- Na rozdíl od lineárního prohledávání zde jako metodu použijeme druhou hashovací funkci
- Obě jsou funkcí k
- Každá má jinou sekvenci prohledávání

$$h(k) = [h_1(k) + i \cdot h_2(k)] \bmod m$$

$$h(k) = [(k \bmod 5) + i \cdot h_2(k)] \bmod 5, \quad h(k) = (k + i \cdot 3) \bmod 5$$

posloupnost: 1 5 **21** 10 7

0	5	
1	1	
2		
3		
4	21	

Zjednodušení, stačí prvočíslo $\neq m$

kolize - 1 blokuje

=> 2. double hashing

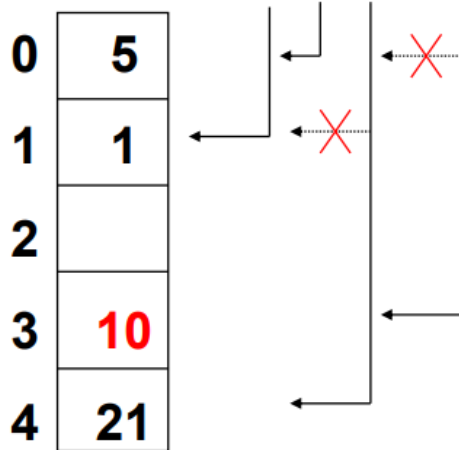
vlož o 3 pozice dál ($i++ \Rightarrow i = 1$)

(i je číslo pokusu)

Double hashing

$$h(k) = (k + i \cdot 3) \bmod 5 \quad (\text{Př. velmi zjednodušené } h(k))$$

posloupnost: 1 5 21 10 7

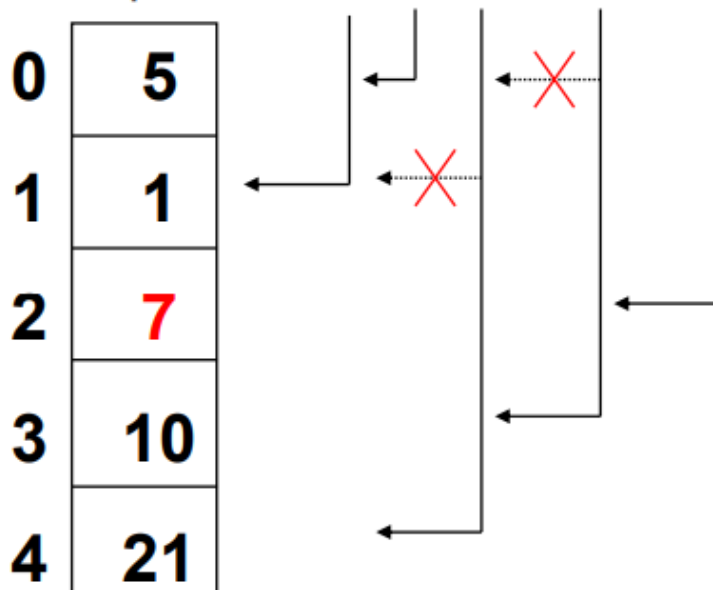


kolize - 5 blokuje - vlož dál

(vlož o 3 pozice dál ($i = 1$))

$$h(k) = (k + i \cdot 3) \bmod 5 \quad (\text{Př. velmi zjednodušené } h(k))$$

posloupnost: 1 5 21 10 7



$i = 0$

$h(k) = (k + i \cdot 3) \bmod 5$ (Př. velmi zjednodušené $h(k)$)

posloupnost: 1 5 21 10 7

0	5	$i = 0$
1	1	$i = 0$
2	7	$i = 0$
3	10	$i = 1$
4	21	$i = 1$