

„hasorání“

30. 10. 2010

$$x = (x_1, x_2, x_3)$$

$$A = (A_1, A_2, A_3)$$

$$F(x) = \sum_{i=1}^3 A_i x_i \text{ MOD } M$$

1

$\left[ \frac{N}{M} \right]$  = ~~nejednotlivější~~ počet prvků v jednom  
řetězku - nejednotlivější stěr

$$\left[ \frac{N}{M} \right] \left[ \frac{N}{M} - 1 \right] M = N - M$$

prvky 1-3 jsou nevhodné = malý rozdíl  
mohu rozšířit na

$$A = (A_1, A_2, \dots, A_{20})$$

po zpracování prvků to se zdá další to...

$$x = (x_1, x_2, x_3) \ 0, 0, 0, \dots \quad \text{další bereme jako 0}$$

$$A = (A_1, A_2, \dots, A_{20}, \dots, A_{25})$$

stále platí funkce  $F(x)$  a výsledek to  
téměř neovlivní

=> takto lze haschovat libovolně dlouhé  
řetězky

Ex. 4.5

2

Ex. 4.1

$M = \{ \text{dvoje intenzit} \}$

$M = \{ (1,2), (2,1), (1,3), (4,1) \}$

Ověřte každou tabulku, zda je relace symetrická  
= pokud je  $(1,2)$ , pak ~~musí~~ musí být i  $(2,1)$  atd.

jak bude  $F(x)$  vypadat?

- výsledek - měl by být FALSE, protože ~~chybí~~ <sup>není</sup>  $(3,1)$
- výsledek pouze TRUE, FALSE
- máme  $\text{ADD}(\text{KEY}, \text{VALUE})$  nebo  $\text{ADD}(\text{VALUE})$   
 $\text{FIND}(\text{KEY})$   
 $\text{REMOVE}(\text{KEY}, \text{VALUE})$



PŘ

3

~~průběh~~

předpoklad - máme velký soubor obsahující  
(transakce, cena, zákazník)

napište algoritmus účkové ceny pro každého  
zákazníka. (\*)

Ex. 4.4.

Kypřetlele realizaci operace "for all"

= dva for cykly s sobě

první - projede pole

druhý - projede prvky v řetězku

(\*)

MAP < customer, integer > #

NEW HashMap < customer, Integer >

for (trans T : LIST) {

INT PRICE = T.GETPRICE();

CUSTOMER C = T.CUSTOMER();

IF (M.CONTAINSKEY()) {

M.PUT (C.PRICE + M.GET(C)) } //if

~~ELSE M.PUT (C)~~

ELSE M.PUT (C, PRICE) } //for