

HASHOVÁNÍ

Karel Horák, Petr Ryšavý

25. května 2016

Katedra počítačů, FEL, ČVUT

Hashovací¹ funkce

1. převádí adresu daného prvku na jemu příslušný klíč
2. vrací pro každý klíč jedinečnou hodnotu
3. pro daný klíč vypočte adresu
4. vrací pro dva stejné klíče různou hodnotu

¹Česky též rozptylovací

Kolize u hashovací (rozptylovací) funkce $h(k)$

1. je situace, kdy pro dva různé klíče k vrátí $h(k)$ stejnou hodnotu
2. je situace, kdy pro dva stejné klíče k vrátí $h(k)$ různou hodnotu
3. je situace, kdy funkce $h(k)$ při výpočtu havaruje
4. je situace, kdy v otevřeném rozptylování dojde dynamická paměť

Navrhněte vhodnou hashovací funkci pro:

1. číslo typu `int`
2. číslo typu `double`
3. dvě čísla typu `int`
4. objekt typu `String`
5. `ArrayList`

Kolize při vkládání klíče do rozptylovací tabulky s otevřeným rozptylováním znamená, že

1. klíč nebude možno do tabulky vložit
2. klíč bude možno do tabulky vložit po jejím zvětšení
3. místo pro klíč v poli je již obsazeno jiným klíčem
4. v paměti není dostatek místa pro zvětšení tabulky
5. kapacita tabulky je vyčerpána

Metoda otevřeného rozptylování

1. generuje vzájemně disjunktní řetězce synonym
2. dokáže uložit pouze předem známý počet klíčů
3. zamezuje vytváření dlouhých clusterů ukládáním synonym do samostatných seznamů v dynamické paměti
4. dokáže uložit libovolný předem neznámý počet klíčů

Do nejprve prázdné tabulky s rozptylovací funkcí $h(x) = x \bmod 6$ byly vloženy následující prvky v uvedeném pořadí a celkem nastala jedna kolize. Která trojice tomuto scénáři vyhovuje?

1. 6 12 24
2. 24 6 12
3. 1 7 6
4. 5 6 7
5. 2 3 4

Double hashing

1. má stejnou pravděpodobnost vzniku dlouhých clusterů jako linear probing
2. je metoda ukládání klíčů na dvě různá místa
3. je metoda minimalizace délky clusterů u metody otevřeného rozptylování
4. má vyšší pravděpodobnost vzniku dlouhých clusterů než linear probing

Metoda hashování s vnějším zřetězením

1. nemá problém s kolizemi, protože při ní nevznikají
2. dokáže uložit pouze předem známý počet klíčů
3. ukládá synonyma do samostatných seznamů v dynamické paměti
4. ukládá synonyma spolu s ostatními klíči v poli
5. ukládá posloupnosti synonym v souvislém úseku adres

Rozptylovací tabulka o velikosti m se zřetězeným rozptylováním obsahuje n prvků. Složitost nejhoršího případu, který může při vložení dalšího klíče nastat, je

1. $\Theta(n)$
2. $\Theta(m)$
3. $\Theta\left(\frac{m}{n}\right)$
4. $\mathcal{O}(1)$
5. $\Theta(\log(n))$

Pro danou rozptylovací funkci $h(k) = k \bmod 5$ zvolte velikost tabulky a nakreslete stav po vložení prvků následující posloupnosti při vnějším zřetězení prvků.

20 9 0 17 22 15 23 18 8 7

Vnější zřetězení: Mohou být prvky s klíči 5 a 15 ve stejném řetězu synonym? Hashovací funkce má tvar $h(k) = k \pmod{13}$.

Do rozptylovací tabulky velikosti 11 s otevřeným rozptylováním a s rozptylovací funkcí $h(k) = k \bmod 8$ vložte následujících 6 klíčů. Použijte strategii Linear Probing s inkrementem 3.

10 16 15 31 23 14

Je nějaký rozdíl v asymptotické rychlosti operace `delete()` ve zřetězeném a otevřeném rozptylování? Jaký nejhorší případ může nastat?

Je nějaký rozdíl v asymptotické rychlosti operace `insert()` ve zřetězeném a otevřeném rozptylování? Jaký nejhorší případ může nastat?

Uložte dané klíče v daném pořadí postupně do rozptylovací tabulky o velikosti 8 s rozptylovací funkcí $h(k) = k \bmod 8$. Použijte strategii LICH a přidaný „sklep tabulky“ o velikosti 2. Určete počet kolizí.

10 12 20 23 32 39 40

Uložte dané klíče v daném pořadí postupně do rozptylovací tabulky o velikosti 7 s rozptylovací funkcí $h(k) = k \bmod 7$. Použijte strategii EICH a přidaný „sklep tabulky“ o velikosti 2. Určete počet kolizí.

9 11 18 27 29 36 43 45