HASHOVÁNÍ

Karel Horák, Petr Ryšavý

25. května 2016

Katedra počítačů, FEL, ČVUT

Příklad 1

Hashovací[1] 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

[1] Česky též rozptylovací

Řešení 1

Správně je 3. odpověď. Hashovací funkce nám říká, ,,kam máme umístit daný prvek" — tj. na jakou adresu ho máme vložit.

Příklad 2

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

Řešení 2

Ke kolizi dochází, když nám hashovací funkce říká, ať vložíme různé prvky na stejnou pozici. Správně je tedy odpověď 1.

Příklad 3

Navrhněte vhodnou hashovací funkci pro:

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

Řešení 3

1. číslo samotné
2. převedeme `double` na `long` a vyxorujeme horní polovinu čísla s dolní
3. `a*p + b`, kde `p` je prvočíslo
4. `((a1*p + a2)*p + a3)*p + a4` ...

Příklad 4

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

Řešení 4

Kolizí při vkládání klíče do hashovací tabulky myslíme, když nám hashovací funkce říká, ať prvek vložíme na pozici, která už je obsazená. Správně je proto odpověď 3.

Příklad 5

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

Řešení 5

Při otevřeném hashování ukládáme prvky do pole o pevně dané velikosti — dokážeme proto uložit pouze předem známý počet klíčů. Pokud by nám už tabulka nestačila, museli bychom vytvořit novou tabulku a všechny prvky do této nové tabulky vložit po jednom.

Příklad 6

Do nejprve prázdné tabulky s rozptylovací funkcí $h(x) = x \mod 6$ 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`

Řešení 6

Správně je odpověď 3. Stav tabulky v jednotlivých fázích je následující:

```
_ _ _ _ _ _
_ 1 _ _ _ _
_ 1 7 _ _ _
6 1 7 _ _ _
```

Příklad 7

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

Řešení 7

Správně je možnost 3. Double hashing se snaží minimalizovat délku clusterů tím, že clusterům přiřazuje různé ,,mezery". Je tak méně pravděpodobné, že budou clustery srůstat.

Příklad 8

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

Řešení 8

Správně je odpověď 3. Pro každou adresu v hashovací tabulce je vytvořený spojový seznam synonym.

Příklad 9

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. $\Theta(\log(n))$

Řešení 9

Pro každou adresu máme k dispozici spojový seznam synonym. Do tohoto seznamu vložíme nový prvek na začátek v čase $\mathcal{O}(1)$.

Příklad 10

Pro danou rozptylovací funkci $h(k) = k \mod 5$ 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

Řešení 10

Zvolme velikost tabulky 5.

```
0 -> 15 0 20
1 ->
2 -> 7 22 17
3 -> 8 18 23
4 -> 9
```

Příklad 11

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

Řešení 11

Nemohou. $h(5) = 5$, $h(15) = 2$.

Příklad 12

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

10 16 15 31 23 14

Řešení 12

```
__ __ __ __ __ __ __ __ __ __ __
__ __ 10 __ __ __ __ __ __ __ __
16 __ 10 __ __ __ __ __ __ __ __
16 __ 10 __ __ __ __ 15 __ __ __
16 __ 10 __ __ __ __ 15 __ __ 31
16 __ 10 __ __ 23 14 15 __ __ 31
```

Příklad 13

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?

Řešení 13

Operace `delete()` v případě zřetězeného rozptylování může trvat až $\mathcal{O}(n)$ — tento čas odpovídá nalezení prvku v seznamu. Jeho vlastní odstranění je pak v konstantním čase.

V případě otevřeného hashování je `delete()` komplikovanější. Nelze jednoduše vyjmout prvek z prostředku clusteru ($\sim$ není to spojový seznam). Můžeme buď přehashovat celý cluster (to je ale drahé), proto se v praxi častěji prvek pouze označí jako „smazaný".

Řešení 13

V zřetězeném hashování je `insert()` v konstantním čase (vlož na začátek ve spojovém seznamu synonym).

V otevřeném hashování musíme najít konec clusteru, složitost proto je $\mathcal{O}(n)$.

Příklad 14

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

10 12 20 23 32 39 40

Řešení 14

Obsah tabulky bude postupně následující:

		10							
		10		12					
		10		12					20
		10		12			23		20
32		10		12			23		20
32		10		12			23	39	20
32		10		12		40	23	39	20

Během vkládání dojde ke 3 kolizím. Prvek 32 odkazuje 40, 12 odkazuje 20 a 23 odkazuje na 39.

Příklad 15

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

9 11 18 27 29 36 43 45

Řešení 15

Obsah tabulky bude postupně následující:

		9						
		9		11				
		9		11				18
		9		11		27		18
	29	9		11		27		18
	29	9		11		27	36	18
	29	9		11	43	27	36	18
	29	9	45	11	43	27	36	18

Během vkládání dojde ke 3 kolizím. Prvek 29 odkazuje 43, 43 odkazuje 30 a 11 odkazuje na 18.