

1.

Metoda hashování s vnějším zřetězením

- a) nemá problém s kolizemi, protože při ní nevznikají
- b) dokáže uložit pouze předem známý počet klíčů
- c) ukládá synonyma do samostatných seznamů v dynamické paměti
- d) ukládá synonyma spolu s ostatními klíči v poli

Každý alespoň elementární popis zřetězeného rozptylování vede na odpověď b). Kolize vznikají vždy, pole se tu nepoužívá a počet klíčů není teoreticky omezen.

2.

Metoda otevřeného rozptylování

- a) generuje vzájemně disjunkttní řetězce synonym
- b) dokáže uložit pouze předem známý počet klíčů
- c) zamezuje vytváření dlouhých clusterů ukládáním synonym do samostatných seznamů v dynamické paměti
- d) dokáže uložit libovolný předem neznámý počet klíčů

Varianty a), c), d) platí zřejmě pro zřetězené rozptylování, což vyplývá bezprostředně již z jakéhokoli jednoduchého popisu zřetězeného rozptylování. Zbývá jen správná možnost b).

3.

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

- a) je situace, kdy pro dva různé klíče k vrátí $h(k)$ stejnou hodnotu
- b) je situace, kdy pro dva stejné klíče k vrátí $h(k)$ různou hodnotu
- c) je situace, kdy funkce $h(k)$ při výpočtu havaruje
- d) je situace, kdy v otevřeném rozptylování dojde dynamická paměť

Definitorická otázka, viz přednášky/literaturu. Platí varianta a).

4.

Cluster (u metody otevřeného rozptylování)

- a) je posloupnost synonym uložená v souvislém úseku adres
- b) je posloupnost klíčů uložená v souvislém úseku adres
- c) je posloupnost synonym uložená v dynamické paměti
- d) u otevřeného rozptylování nevzniká

Definitorická otázka, viz přednášky/literaturu. Platí varianta a).

5.

Metoda hashování s vnějším zřetězením

- a) nemá problém s kolizemi, protože při ní nevznikají
- b) dokáže uložit pouze předem známý počet klíčů
- c) ukládá synonyma do samostatných seznamů v dynamické paměti
- d) ukládá synonyma spolu s ostatními klíči v poli

Každý alespoň elementární popis zřetězeného rozptylování vede na odpověď b). Kolize vznikají vždy, pole se tu nepoužívá a počet klíčů není teoreticky omezen.

6.

Double hashing

- a) má stejnou pravděpodobnost vzniku dlouhých clusterů jako linear probing
- b) je metoda ukládání klíčů na dvě různá místa
- c) je metoda minimalizace délky clusterů u metody otevřeného rozptylování
- d) má vyšší pravděpodobnost vzniku dlouhých clusterů než linear probing

Double hashing je vymyšlen právě proto, aby v otevřeném rozptylování pokud možno nevznikaly dlouhé souvislé posloupnosti synonym (= clustery). Možnosti a) a d) neplatí, možnost b) je vůbec beze smyslu, zbývá jen korektní možnost c).

7.

Double hashing

- a) je metoda ukládání klíčů na dvě různá místa současně
- b) je metoda minimalizace kolizí u metody otevřeného rozptylování
- c) má vyšší pravděpodobnost vzniku kolizí než linear probing
- d) je metoda minimalizace kolizí u metody rozptylování s vnějším zřetězením

Tady nepomůže asi nic jiného než dobrá paměť.

8.

Hashovací (=rozptylovací) funkce

- a) převádí adresu daného prvku na jemu příslušný klíč
- b) vrací pro každý klíč jedinečnou hodnotu
- c) pro daný klíč vypočte adresu
- d) vrací pro dva stejné klíče různou hodnotu

Toto je elementární otázka z rozptylování. Hashovací funkce podle své definice provádí činnost popsanou ve variantě c). Vytváří synonyma, takže varianta b) neplatí, a pro dva stejné klíče musí vrátit stejnou hodnotu, takže ani d) neplatí. Varianta a) obsahuje jen klíčová slova beze smyslu naházená do věty — neplatí také.

9.

V otevřeném rozptylování

- a) je nutno definovat rozsah hodnot klíčů
- b) je počet uložených prvků omezen velikostí pole
- c) je nutno po určitém počtu kolizí zvětšit velikost pole
- d) je možno uložit libovolný počet synonym

V otevřeném rozptylování je maximální počet uložených prvků dán velikostí pole, varianta d) neplatí. Většinou se počítá s tím, že pole má danou velikost (podle charakteru a rozsahu dat), jeho velikost je tedy daná a nemění se. Varianta c) neplatí. Zároveň se potvrzuje platnost varianty b). Varianta a) neplatí, rozptylovací funkce má za úkol zpracovat jakýkoli klíč.

10.

Zřetězený seznam synonym

- a) minimalizuje délku clusterů u metody otevřeného rozptylování

- b) u otevřeného rozptylování nevzniká
- c) je posloupnost synonym uložená v souvislém úseku adres
- d) řeší kolize uložením klíče na první volné místo v poli

Otázka opět spíše definitorická. Zřetězený seznam dává jméno zřetěženému rozptylování, takže se v otevřeném rozptylování neobjevuje. Tím je vyloučena varianta a). Je nutná určitá představa o otevřeném rozptylování, neboť formulace variant c) a d) naznačuje, že se týkají právě jeho. Zbývá tak jen zřejmá správná varianta b)

11.

Pole, ve kterém je uložena rozptylovací tabulka vypadá při použití rozptylovací funkce $h(k) = k \bmod 5$, lineárního prohledávání (linear probing) a vložení klíčů 6, 4, 5, 9 (vkládaných v pořadí zleva doprava) takto

a)

0	1	2	3	4
5	6	4		9

b)

0	1	2	3	4
5	6	9		4

c)

0	1	2	3	4
4	6	5		9

d)

0	1	2	3	4
4	5	6		9

$6 \bmod 5 = 1$, takže hodnota 6 se vloží na pozici č. 1.

$4 \bmod 5 = 4$, takže hodnota 4 se vloží na pozici č. 4.

$5 \bmod 5 = 0$, takže hodnota 5 se vloží na pozici č. 0.

Tím získáme tabulku naznačenou vpravo.

Zbývá vložit hodnotu 9.

$9 \bmod 5 = 4$. Hodnota 9 by se tedy měla vložit na pozici č. 4. Ta je však již obsazená hodnotou 4 a je tedy zapotřebí najít nejbližší volnou pozici směrem doprava. Za pozicí č. 4 bezprostředně následuje pozice č. 0.

(tabulka je „zacyklená“) a pozice č. 1, jež jsou obě také obsazeny,

Hodnota 9 se tedy vloží na pozici č. 2, jak ukazuje poslední tabulka, ekvivalentní s variantou b).

0	1	2	3	4
5	6			4

0	1	2	3	4
5	6	9		4

12.

Pole, ve kterém je uložena rozptylovací tabulka vypadá při použití rozptylovací funkce $h(k) = k \bmod 5$, lineárního prohledávání (linear probing) a vložení klíčů 6, 5, 9, 4 (vkládaných v pořadí zleva doprava) takto

a)

0	1	2	3	4
5	6	4		9

b)

0	1	2	3	4
5	6	9		4

c)

0	1	2	3	4
4	6	5		9

d)

0	1	2	3	4
4	5	6		9

$6 \bmod 5 = 1$, takže hodnota 6 se vloží na pozici č. 1.

$5 \bmod 5 = 0$, takže hodnota 5 se vloží na pozici č. 0.

$9 \bmod 5 = 4$, takže hodnota 9 se vloží na pozici č. 4.

Tím získáme tabulku naznačenou vpravo.

Zbývá vložit hodnotu 4.

$4 \bmod 5 = 4$. Hodnota 4 by se tedy měla vložit na pozici č. 4. Ta je však již obsazená hodnotou 9 a je tedy zapotřebí najít nejbližší volnou pozici směrem doprava. Za pozicí č. 4 bezprostředně následuje pozice č. 0.

(tabulka je „zacyklená“) a pozice č. 1, jež jsou obě také obsazeny,

Hodnota 4 se tedy vloží na pozici č. 2, jak ukazuje poslední tabulka, ekvivalentní s variantou a).

0	1	2	3	4
5	6			9

0	1	2	3	4
5	6	4		9

13.

Implementujte operace Init, Search, Insert a Delete pro rozptylovací tabulku se zřetěženým rozptylováním, do níž se ukládají celočíselné klíče. Předpokládejte, že rozptylovací funkce je již implementována a Vám stačí ji jen volat.

14.

Implementujte operace Init, Search, Insert pro rozptylovací tabulku s otevřeným rozptylováním, do níž se ukládají celočíselné klíče. Předpokládejte, že rozptylovací funkce je již implementována a Vám stačí ji jen volat. Použijte strategii „Linear probing“.

15.

Implementujte metodu hashování s otevřeným rozptylováním a hashovací funkcí $h(x) = k \bmod 691$. Kolize řešte metodou linear probing.

Zde (13., 14., 15) – pokud se nevyskytne přímá žádost – řešení prozatím neuvádíme, jedná se jen o přímou implementaci standardní situace popsané v přednášce i literatuře, nic se tu nemusí „vymýšlet“, předpokládáme tedy, že si zájemci mohou (příp. s knihou či obrazovkou) tamtéž uvedené kódy projít.