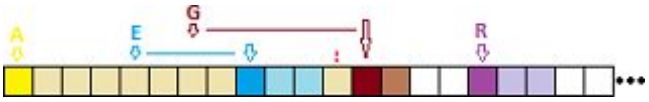
Rozptylové tabulky s s vnějším řetězením

Z FAV wiki

HASH Pokud máme množinu záznamů, ve které mohou v klíčích existovat duplicity pro různé hodnoty (například tabulka slov podle prvího písmena). Pro tento přístup je nemožné používat přímé adresování, jelikož by docházelo ke kolizím.



Můžeme ale v poli využít prázdná místa a do nich duplicitní hodnoty ukládat. Vyhladávání a ukládání potom obecně nebude v konstantním čase. Můžeme tedy například uložit duplicitní hodnotu do následujícího prázdného místa v tabulce. Problémy jsou však následující:

Můžeme zabrat místo pro klíč, který má na toto místo právoplatný nárok (kolize tedy nastanou nejen pro hodnoty se stejným klíčem, ale i obecně pro libovolný klíč) Odsuneme-li i tento klíč, dojde k vytváření shluků (a nabalování dalších klíču na tyto shluky, na obr. skluk od písmene A) Dochází také k fragmentaci, je tedy nutné hledat až do prvního prázdného klíče, což je při shlukování a vyšší saturaci tabulky pomalé

Tento přístup tedy použijeme, pokud vímě, že jsou klíče hodnot rovnoměrně rozmístěny, a že hodnot není více než klíčů, a nazývá se vnitřní zřetězení.

Vnější zřetězení znamená, že ke kažnému klíči přiřadíme seznam hodnot. Nedochází tedy ke kolizím, ale vyhledání hodnoty je O(m), kde m je počet hodnot ke každému klíči, tedy nejhůže O(n) (všechny hodnoty jsou v jednom klíči). Tento způsob však zvládá i situace, kde je více hodnot než klíčů

Rozptýlení klíčů.

Pro vnitřní zřetezení lze použít lepší vyhledáváci (a ukládací) funkci, než lineární posun, třeba kvadratickou fci. Kvadratická funkce se může protonout s jinou kv. fcí s jiným počátkem jen jednou) Tím omezíme clustery a kolize, a ukládání a hledání je rychlejší.

Pro vnější zřetezení, nebo pokud je při vnitřním zřetězení mnoho podobných klíčů, je vhodné použít orákulum, rozptylovou (hashovací) funkci. Ta má 2 vlastnosti:

Namapuje hodnoty do rozsahu počtu klíčů Ideálně zruší nebo alespoň naruší vztahy mezi podobnými klíči

Příklad: Slovník se vnitřním zřetězením. Slovo převedeme na číslo součtem ASCII hodnot znaků

Vlastnost 1: na toto číslo použijeme operaci modulo. Tato operace je pomalá (jde o dělení), použijeme tedy ideálně velikosti tabulky, které jsou mocninou 2, a po té je dělení pouhým bitovým posunem Vlastnost 2: vazby v tomto postupu existují (například anagramy se namapují do stejného pole). Použijeme tedy například následující postup: (((1. písmeno * (2k+1)) + 2. písmeno) * (2k+1)) + ... Tím zrušíme vazby mezi podobnými slovy. (2k+1) použijeme, protože jde o bitový posun a přičtení 1, což je velmi rychlé a 1 přičítáme, protože by poté modulo ztratilo význam.

Mohli bychom použít i kdyptografické hashování, jako MD5, to je však pro O(1) přístup k seznamům příliš pomalé.