Tablice mieszające (haszujące) – hash table

1.**Tablica asocjacyjna** - abstrakcyjny typ danych do przechowywania par **<klucz, dane>**. Umożliwia dostęp do wartości poprzez podanie klucza. Typowe operacje na tablicy asocjacyjnej:
a) czy dany klucz istnieje w tablicy (bez pobierania danych)
b) dodawanie klucza z danymi
c) usunięcie klucza wraz z odpowiadającymi im danymi

2.Tablica mieszające jako sposób implementacji tablicy asocjacyjnej

3.Tablice z adresowaniem bezpośrednim – tyle indeksów ile jest możliwych kluczy, wykorzystujemy klucz jako indeks tablicy (w przypadku kluczy nie liczbowych trzeba je zamienić na wartość liczbową)

Wady: Rozmiar tablicy jest związany z ilością możliwych kluczy, a nie z ilością rzeczywistych elementów->dużo pamięci

Tablice mieszające (haszujące) – hash table

U = zbiór wszystkich możliwych kluczy (słów)
S = aktualna zawartość słownika , $S \subseteq U$, $|S| = n$ (ilość elementów słownika)

- elementy zbioru S przechowywane w tablicy mieszającej: A[0 .. m-1]
- wartość klucza służy do obliczenia adresu (indeksu tablicy), pod którym ma się znajdować element
- z tablicą skojarzona jest na stałe funkcja (funkcja mieszająca) odwzorowująca klucze(słowa) w adresy (indeksy tablicy)

$h : U \to \{0, . . . , m-1\}$

Współczynnik zapełnienia tablicy
α = n/m gdzie n – ilość elementów w tablicy, m – pojemność
Wymagania stawiane funkcjom mieszającym, osiągalne tylko w przybliżeniu:

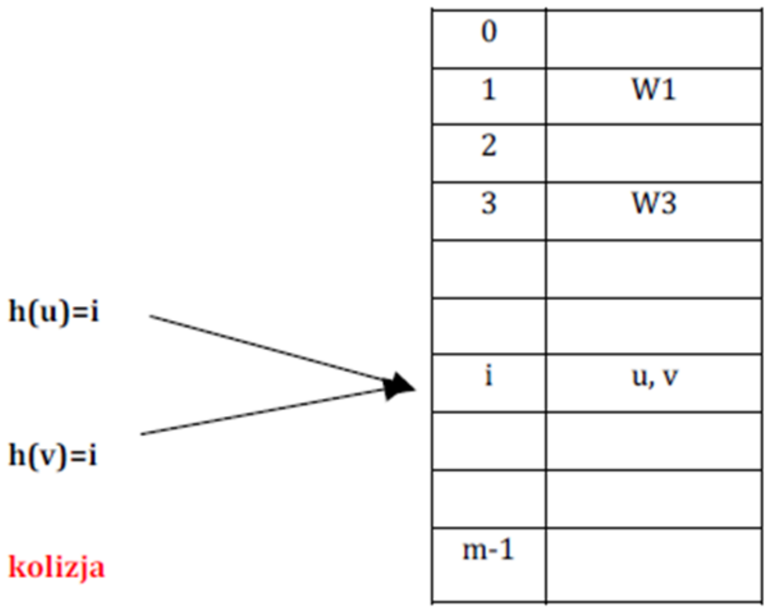
- łatwo obliczalna
- losowa, tzn. każdy indeks jednakowo prawdopodobny jako wartość funkcji dla losowego klucza, niezależnie od wartości dla innych kluczy znajdujących się w S (to założenie nazywa się prostym mieszaniem jednostajnym)

Tablice mieszające (haszujące) – hash table

Najprostsza funkcja haszująca - funkcja modularna

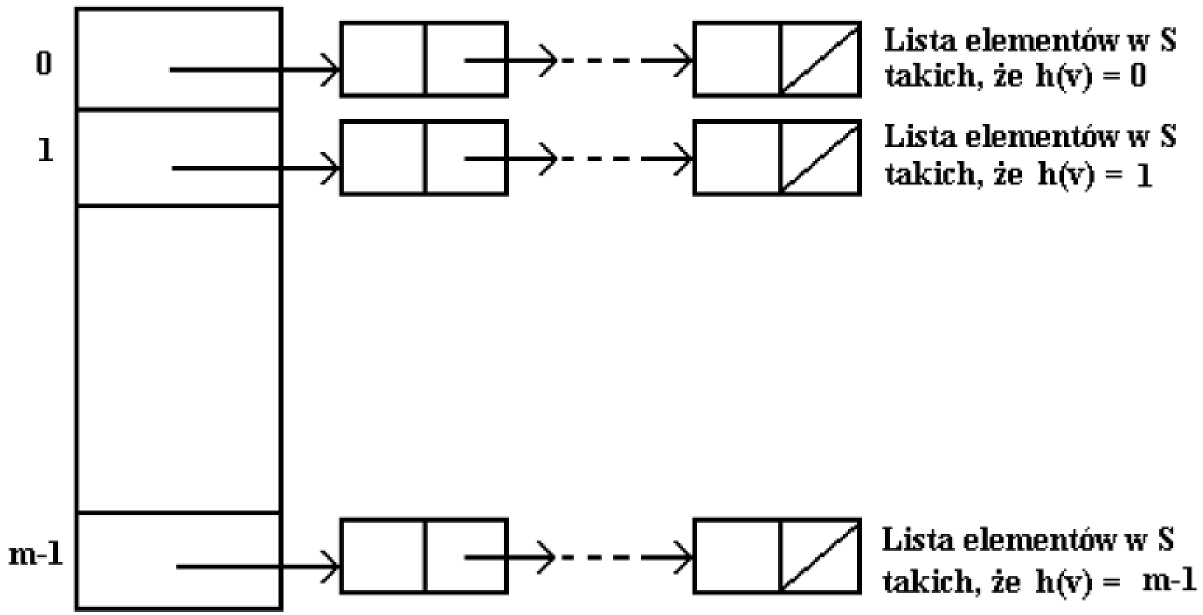
- zamiana wartości *key* na liczbę *num* (jeśli *key* nie jest liczbą)
- funkcja modulo m h(*num*) = *num*% m

Ponieważ najczęściej |U| >> m to może się zdarzyć, że h(u)=h(v) <= KOLIZJA



Metody rozwiązywania kolizji

I. Metoda łańcuchowa



Tablice mieszające (haszujące) – hash table (2)

Metoda łańcuchowa (c.d.)

Search(v)
Złożoność:
1. pesymistyczna O(n) (wszystkie elementy zbioru S, n=|S| mogły trafić do jednej listy)
2. średnia, przy założeniu, że funkcja mieszająca daje proste mieszanie jednostajne:
w przypadku sukcesu równa połowie oczekiwanej długości listy, a więc O(n/m).
w przypadku porażki równa oczekiwanej długości listy, a więc O(n/m).

Insert(v)
Złożoność: O(1)

Delete(v)
Złożoność jak w operacji Search(v).
Często operacja Delete(v) nie usuwa fizycznie elementu v, lecz jedynie markuje, że go nie ma słowniku, zmieniając klucz v na nonItem.

II. Adresowanie otwarte (założenie: $n < m$)

W metodzie adresowania otwartego wszystkie elementy, również kolidujące, umieszczane są w tablicy rozproszonej. Gdy element danych nie może zostać umieszczony na pozycji w tablicy, która odpowiada wyliczonej przez funkcję rozpraszającą wartości indeksu, wyszukiwana jest inna wolna pozycja. W przypadku kolizji mieszanie wyznacza ciąg prób: ciąg indeksów, w których kolejno sprawdza się zawartość tablicy.

$$h : U \to \{0,1,...,m-1\} \times \{0,1, \ldots ,m-1\}$$

przy czym wymagane jest aby ciąg prób: h(key, 0), h(key, 1), ... , h(key, m-1) był permutacją liczb $\{0,1, \ldots ,m-1\}$.

Adresowanie otwarte – funkcja liniowa

Ciąg prób:

h(key, i) = (h'(key) + i) % m, i=0,1, ... ,m-1

Działanie tej metody opiera się na założeniu, że tablica nie może być całkowicie wypełniona (istnieje pusta komórka), w przeciwnym przypadku metoda search() może wpaść w nieskończoną pętlę.

Niech $\alpha = n/m$ współczynnik zapełnienia, założenie: $\alpha < 1$

Średnia liczba porównań (miejsc przeglądanych):

- wyszukiwanie trafione: $= 0.5 (1 + 1/(1- \alpha))$
- wyszukiwanie chybione (lub wstawianie): $= 0.5 (1 + 1/(1- \alpha)^2)$

Metoda prosta i efektywna gdy $\alpha < 1/2$. W przeciwnym przypadku tworzą się skupiska wypełnionych miejsc w tablicy, opóźniające wyszukiwanie

Adresowanie otwarte – funkcja kwadratowa

Ciąg prób:

$$h(key, i) = (h'(key) + a*i^2 + b*i) \% m$$

Adresowanie kwadratowe eliminuje problem grupowania elementów danych, występujący w metodzie adresowania liniowego. Idea metody polega na tym, aby nie sondować komórek sąsiadujących, ale komórki, które pozostają w pewnym oddaleniu od siebie.

Okazuje się, że wybór stałych a i b innych niż a=1, b=0 nie wnosi nic pożytecznego, czyli obliczenie kolejnych adresów można uprościć:

$$h(key, i) = (h'(key)+i^2) \% m.$$

Wyznaczając adresy według powyższego wzoru można udowodnić, że obejmujemy szukaniem tylko połowę tablicy.

Adresowanie otwarte – podwójna funkcja

Aby wyeliminować zarówno grupowanie pierwotne, jak i grupowanie wtórne stosuje się dodatkową funkcję mieszającą – g.

h(key, i) = (h'(key) + i*g(key)) % m

Badania empiryczne wskazują, że dodatkowa funkcja mieszająca musi spełniać następujące warunki:

- g istotnie różna od h, przykładowa definicja wtórnej g(key): g(key)=p*(key % p), gdzie: p – liczba pierwsza mniejsza od m
- nie może zwracać 0 (wynikiem byłaby nieskończona pętla)
- g(key) - względnie pierwsza z m; przykładowe sytuacje gdy ta własność zachodzi:

1. m jest liczbą pierwszą
2. m jest potęgą 2, a wartości g(key) są nieparzyste. Np. dla funkcji h(key) = key % m , dobre wyniki daje: g(key) = (m–2)* key % (m- 2)

Adresowanie otwarte – podwójna funkcja

UWAGA:
Konieczne jest aby podczas jednego przeszukiwania potencjalnie cała tablica była przeglądana;

Złożoność wyszukiwania:

- pesymistyczna – liniowa.
- średnia, oszacowana dla założenia, że "rozpraszanie jednostajne", tzn. każda permutacja prób jest jednakowo prawdopodobna:
    - wyszukiwanie z sukcesem: $-\ln(1-\alpha)/\alpha$
    - wyszukiwanie z brakiem sukcesu: $1/(1-\alpha)$