



ALGORITMI I STRUKTURE PODATAKA

STUDIJSKI PROGRAMI:

SOFTVERSKO INŽENJERSTVO, RAČUNARSKA TEHNIKA, INFORMATIKA I MATEMATIKA

NASTAVNIK: DOC. DR ULFETA MAROVAC, UMAROVAC@NP.AC.RS

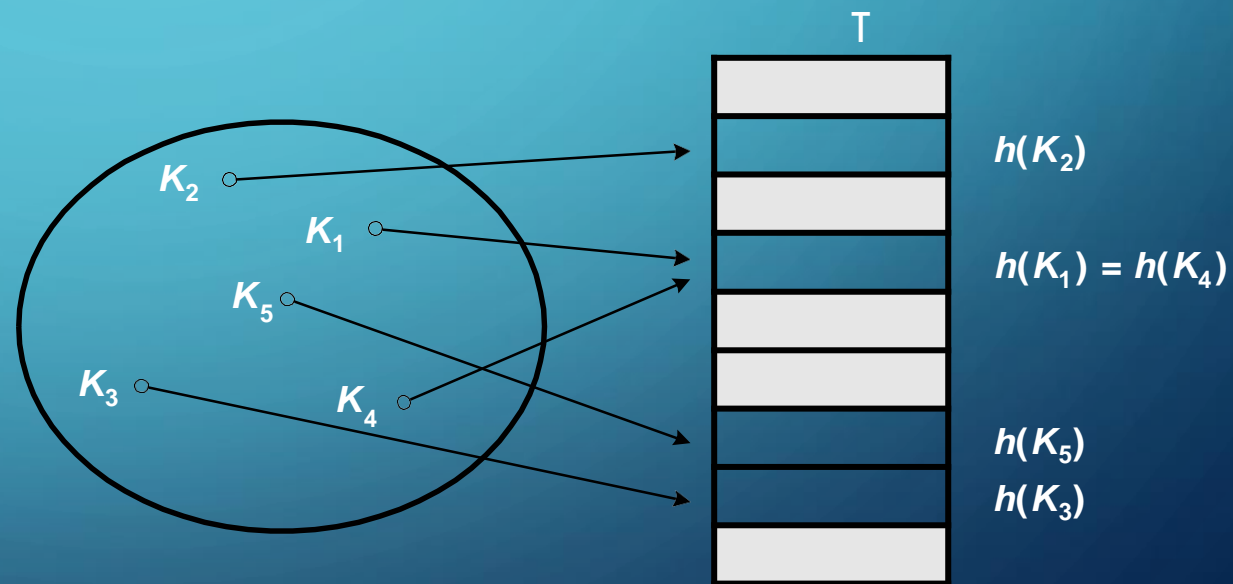
ALGORITMI I STRUKTURE PODATAKA

HEŠIRANJE

- Obično performanse kod pretraživanja zavise od broja ključeva u skupu
- Idealno pretraživanje – na osnovu ključa direktan pristup
 - bez poređenja sa drugim ključevima
- Moguće sa dovoljno velikom tabelom,
 - ali često nepraktično za prirodne ključeve
- *Heš funkcija* mapira skup ključeva
 - na opseg indeksa u *heš tabeli* (samo ključevi ili zapisi)
- Direktno ili rasuto adresiranje

HEŠIRANJE

- heš tabela $T[i]$, $0 \leq i \leq n - 1$
- matična adresa $i = h(K)$



HEŠIRANJE

- Zadatak heš funkcije – kompresija skupa ključeva na manji opseg indeksa tabele
- Kolizija – $h(K_i) = h(K_j), K_i \neq K_j$
 - ✓ sinonimi \Rightarrow klasa ekvivalencije
 - ✓ degradacija performanse
- Poželjne osobine za efikasnost pristupa
 - ✓ uniformnost – $P(i = h(K)) = 1/n, 0 \leq i \leq n - 1$
 - ✓ održavanje poretka – $h(K_i) > h(K_j)$ za $K_i > K_j$
- Izbor
 - ✓ efikasne heš funkcije
 - ✓ efikasnog metoda za razrešavanje kolizija

HEŠ FUNKCIJE

- Kriterijumi za izbor heš funkcije
 - ✓ jednostavnost \Rightarrow brže izračunavanje
 - ✓ uniformnost \Rightarrow ređe kolizije
- Ključevi – numerički, alfanumerički, alfabetski
- Metod ekstrakcije
- Poželjno da heš funkcija zavisi od svih znakova ključa, kao i njihovih pozicija
- Heš funkcije
 - ✓ nezavisne od raspodele ključeva
 - ✓ zavisne od raspodele ključeva

NEZAVISNE HEŠ FUNKCIJE

- Metod deljenja – $h(K) = K \bmod n$
 - ✓ $n \leq$ veličina heš tabele
 - ✓ jednostavan i često korišćen
 - ✓ ne preporučuje se da n bude:
 - parno,
 - stepen broja 2 ili 10
 - neprosto sa modulom kongruencije
 - ✓ u praksi se preporučuje da bude prost broj ne previše blizu stepena broja 2
- Metod množenja - $h(K) = n(cK \bmod 1)$, $0 < c < 1$
 - ✓ izbor n nije kritičan (može i $n = 2^p$)
 - ✓ $c \approx 0.61803$ ("zlatni presek")

NEZAVISNE HEŠ FUNKCIJE

Metod sredine kvadrata

$K = 5894$

	<u>5894 * 5894</u>
	23576
	53046
	47152
	<u>29470</u>
	34739236
	↓
	39

Metod sklapanja

$K = 19653014$

19	19
65	56
30	30
14	41
<u>128</u>	<u>146</u>
↓	↓
28	46
a)	b)

NEZAVISNE HEŠ FUNKCIJE

- Metod konverzije osnove
 - ✓ ključ u sistemu sa osnovom p
 - interpretira se u sistemu sa osnovom q ($q > p$)
 - ✓ bira se q uzajamno prosto sa p
 - ✓ npr. 6154 ($p = 10$) \Rightarrow 13420 ($q = 13$)
- Metod algebarskog kodovanja
 - ✓ ključ $(k_{r-1} \dots k_0) \Rightarrow K(x) = k_{r-1}x^{r-1} + \dots + k_0$
 - ✓ za $n = 2^m$, odabere se $P(x) = x^m + p_{m-1}x^{m-1} + \dots + p_0$
 - ✓ $K(x) \bmod P(x) = h_{m-1}x^{m-1} + \dots + h_0$.
 - ✓ rezultat $(h_{m-1} \dots h_0)$
 - ✓ hardverska implementacija

NEZAVISNE HEŠ FUNKCIJE

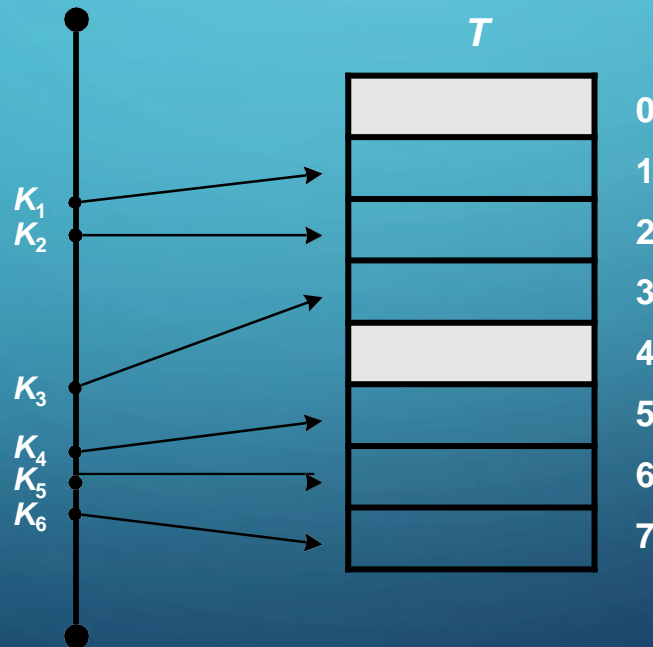
- Savršena heš funkcija
 - ✓ nema kolizija
 - ✓ nije lako pronaći, pogotovo za dinamičan skup
 - ✓ lakše za manje popunjene tabele
 - ✓ npr. $H(K) = (K + s)/d$ (Sprugnoli)
 - ✓ minimalna savršena – n ključeva u n ulaza
 - ✓ npr. $H(K) = c/p(K)$ (Chang)
- Univerzalna klasa heš funkcija
 - ✓ slučajan izbor heš funkcije iz konačnog skupa nezavisan od skupa ključeva
 - ✓ dobre prosečne performanse i zaštita od najgoreg slučaja

ZAVISNE HEŠ FUNKCIJE

- Moguće za unapred poznatu raspodelu ključeva
- Metod analize cifara
- Diskretna kumulativna funkcija raspodele
 - ✓ $F_z(x) = P(Z \leq x)$
 - ✓ ključevi $K_1 < \dots < K_m$
 - ✓ diskretna uniformna raspodela
 - $P(F_z(Z) \leq i/m) = i/m$, za $0 \leq i \leq m$
 - ✓ $F_z(K_1) = 1/m, F_z(K_2) = 2/m, \dots, F_z(K_m) = 1$
 - ✓ cilj $P(h(K) = i) = 1/n$, za $0 \leq i \leq n-1$
 - ✓ $H(K) = \lceil nF_z(K) \rceil - 1$

ZAVISNE HEŠ FUNKCIJE

- Primer:
 - ključevi 6, 7, 12, 14, 15 i 16 u tabeli sa 8 ulaza



RAZREŠAVANJE KOLIZIJA

- Kolizija – dva ili više ključeva ima istu matičnu adresu
- Veća tabela \Rightarrow ređe kolizije,
 - ali dodatni prostor i manja popunjenost
- Faktor popunjenosti
- Varijante razrešavanja kolizija
 - ✓ otvoreno adresiranje – u okviru tabele
 - ✓ ulančavanje – lista sinonima

OTVORENO ADRESIRANJE

- Ispitni niz
 - ✓ niz adresa za ključ pri pretraživanju ili umetanju
 - ✓ generisanje niza – ponovno heširanje
 - ✓ poželjno – permutacija od $(0 \dots n-1)$
- Problem brisanja
 - ✓ “poluslobodne” lokacije
 - ✓ selektivno pomeranje
- Varijante otvorenog adresiranja
 - ✓ linearno pretraživanje
 - ✓ slučajno pretraživanje
 - ✓ kvadratno pretraživanje
 - ✓ dvostruko heširanje

OTVORENO ADRESIRANJE

HASH-INSERT(T, K)

$i = 0$

repeat

$j = h(K)$

if ($T[j] = \text{empty}$) **then**

$T[j] = K$

return j

else

$i = i + 1$

end_if

until $i = n$

ERROR(Tabela puna)

HASH-SEARCH(T, K)

$i = 0$

repeat

$j = h(K)$

if ($T[j] = K$) **then**

return j

else

$i = i + 1$

end_if

until ($T[j] = \text{empty}$) or ($i = n$)

return *empty*

LINEARNO PRETRAŽIVANJE

- $h_i(K) = (h_0(K) + i) \bmod n$, $i = 1, 2, \dots, n - 1$
- $h_i(K) = (h_{i-1}(K) + 1) \bmod n$

T	
0	45
1	10
2	
3	
4	40
5	5
6	
7	
8	26

a)

T	
0	45
1	10
2	35
3	
4	40
5	5
6	13
7	
8	26

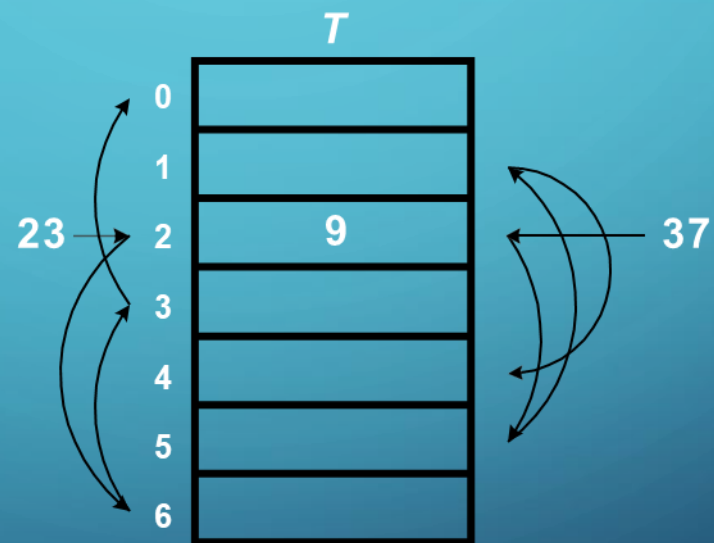
b)

DVOSTRUKO HEŠIRANJE

- $h_i(K) = (h_0(K) + i g(K)) \bmod n, \quad i = 1, 2, \dots, n - 1$
- $h_i(K) = (h_{i-1}(K) + g(K)) \bmod n$
- Za nezavisne $h(K)$ i $g(K)$
verovatnoća sekundarnog grupisanja vrlo mala
- Obično n prost broj, a $g(K) \Rightarrow (1, n - 1)$
- Knuth predlaže
 - ✓ $h_0(K) = K \bmod n$
 - ✓ $g(K) = 1 + K \bmod (n - 2)$
 - ✓ n i $n - 2$ prosti brojevi

DVOSTRUKO HEŠIRANJE

- $h_0(K) = K \bmod 7$
- $g(K) = 1 + K \bmod 5$

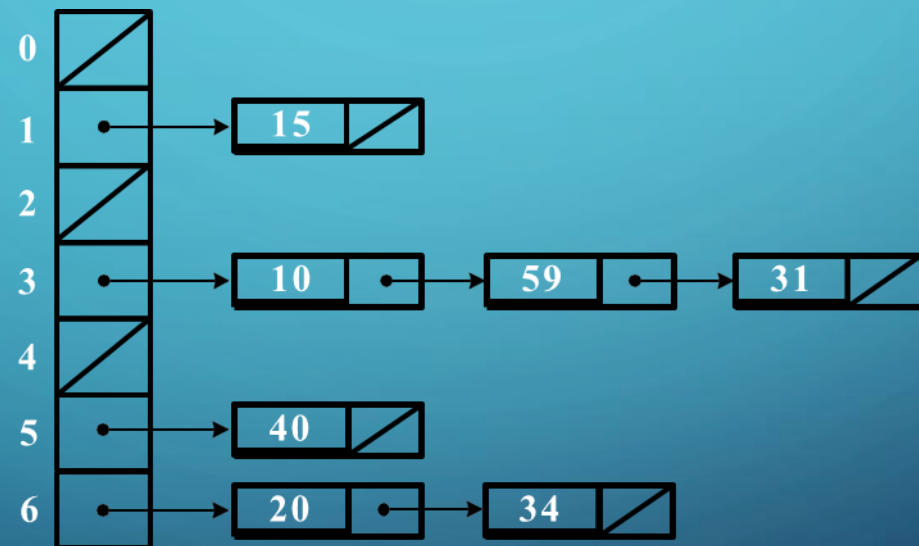


ULANČAVANJE

- Problemi otvorenog adresiranja
 - ✓ brisanje ključeva
 - ✓ ispitni niz ne mora da sadrži samo sinonime
- Ulančavanje sinonima u liste van oblasti tabele
- Lista – klasa ekvivalencije
- Pretraživanje, umetanje i brisanje efikasnije
- Nema ponovnog heširanja
- Ne ograničava broj ključeva veličinom tabele

ODVOJENO ULANČAVANJE

- $h(K) = K \bmod 7$



STANDARDNE TEHNIKE

- Popunjenost $\alpha = m / nb$
- Izbor heš funkcije kritičan – uniformnost!
- Adresiranje baketa – b ključeva bez kolizije
- Ključevi unutar baketa mogu biti i uređeni
 - pa može i binarno pretraživanje
- Neuspešno pretraživanje kada ključ nije u baketu, a on nije pun
- Isti metodi razrešavanja kolizija

SPOLJAŠNJE HEŠIRANJE

- U datotekama sa direktnim pristupom
- Slične heš funkcije i metodi razrešavanja kolizija
- Mnogo bitnije vreme nego prostor
- Cilj – smanjiti broj kolizija
- Ulazi heš tabele kao baketi
 - Baket može sadržati samo ključeve
 - ili cele zapise (nema posebne heš tabele)

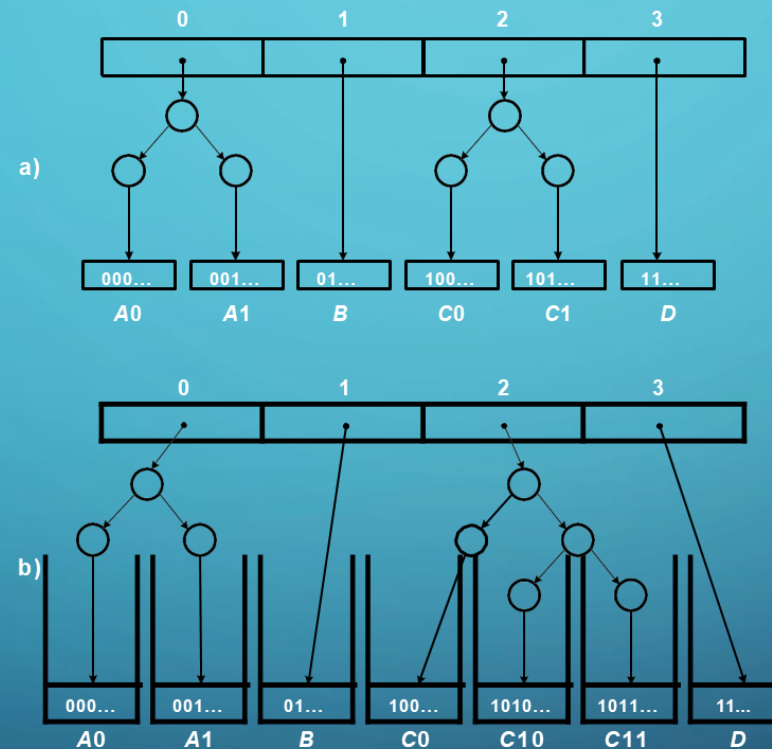
FLEKSIBILNE TEHNIKE

- Datoteke relativno dugovečne strukture
- Veličina može nepredvidljivo da raste
- Standardne tehnike mogu da budu neefikasne
- Fleksibilne tehnike
 - ✓ dinamičko heširanje
 - ✓ proširljivo heširanje

DINAMIČKO HEŠIRANJE

- Primarna oblast m baketa – m ulaza heš tabele
- Heš tabela se adresira sa $b = \log m$ bita
- Umetanje u pun baket izaziva prelom
- Reheširanje i preraspodela po $b + 1$ -vom bitu
- Binarna stabla sa korenima u heš tabeli
- Pretraživanje i brisanje

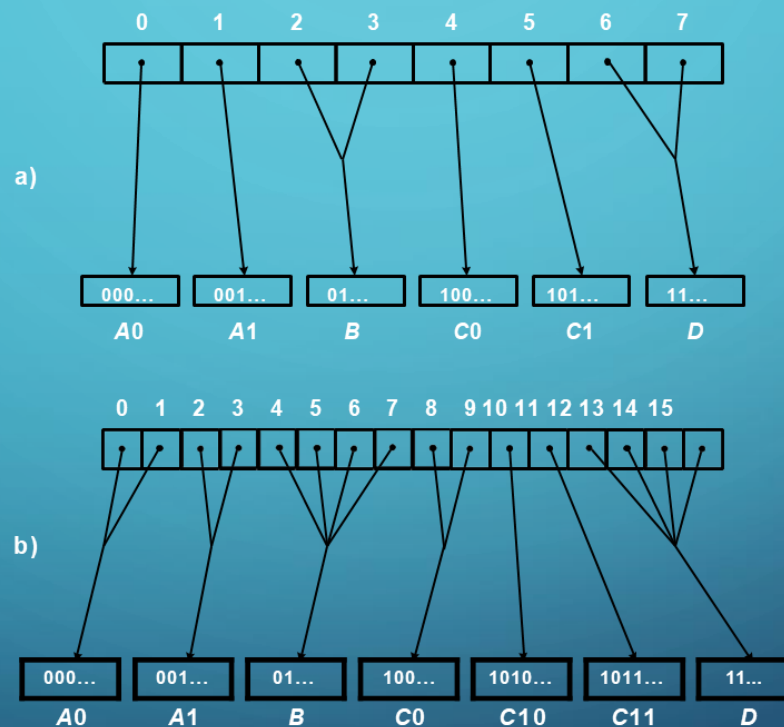
DINAMIČKO HEŠIRANJE



PROŠIRLJIVO HEŠIRANJE

- Prelom i preraspodela kao kod dinamičkog heširanja
- Dubina baketa i dubina heš tabela (d)
- Pri povećanju dubine baketa – razdvajanje ulaza
- Pri povećanju dubine heš tabele – dupliranje
- Heš tabela se adresira sa d bita
- Kad se pri brisanju smanji dubina tabele,
 - tabela se prepolovi

PROŠIRLJIVO HEŠIRANJE



TEST PITANJA

ALGORITMI I STRUKTURE PODATAKA