



ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ
И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Хеширање

АЛГОРИТМИ И

ПОДАТОЧНИ СТРУКТУРИ

- предавања -



Содржина

- ☐ Хеш табели
- ☐ Методи за хеширање
- ☐ Хеш функции
- ☐ Примена

Абстрактен податочен тип хеш табела

- ❑ Обезбедува ефикасни операции за вметнување и пронаоѓање
- ❑ Нема потреба од подредување
- ❑ Не се бара ефикасно бришење
- ❑ Во идеален случај, вметнувањето и пронаоѓањето треба да имаат сложеност од $O(1)$ во најлош случај
 - Но ќе се задоволиме и со $O(1)$ во просечен случај

Можности за имплементација

- ❑ Единствено исплатлива имплементација би била базирана на низа
- ❑ Динамички поврзаните листи не дозволуваат истовремено пристап во $O(1)$ и пребарување во $O(1)$

Принципи на хеширање

- ❑ Да го разгледаме парот (ключ, вредност), наречен уште и **мапа**
- ❑ Ако клучевите во оваа мапа се мали цели броеви, тогаш мапата може да ја претставиме со низа каде клучевите се индексите.
 - Во тој случај, сите операции имат сложеност $O(1)$
- ❑ Идејата е истиот тој пристап да го прошириме за произволен тип на клучеви

Принципи на хеширање

- ❑ **Хеширање**: преведување на произволен клуч во мал цел број кој ќе се користи како индекс во некоја низа
- ❑ **Хеш табела**: низа од m “кофички”, заедно со хеш функција $hash(k)$ која секој клуч k го пресликува во вредност 0 до $m-1$, односно во индексот на неговата “кофичка”

Илустрација

❑ Хеширање

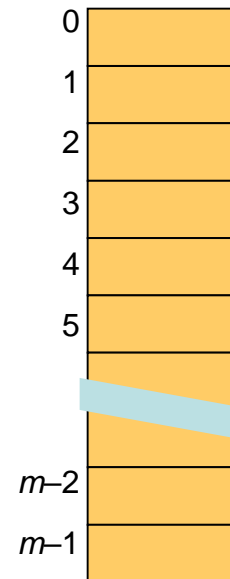
key	value
k_1	v_1
k_2	v_2
k_3	v_3
k_4	v_4
k_n	v_n

Илустрација

Хеширање

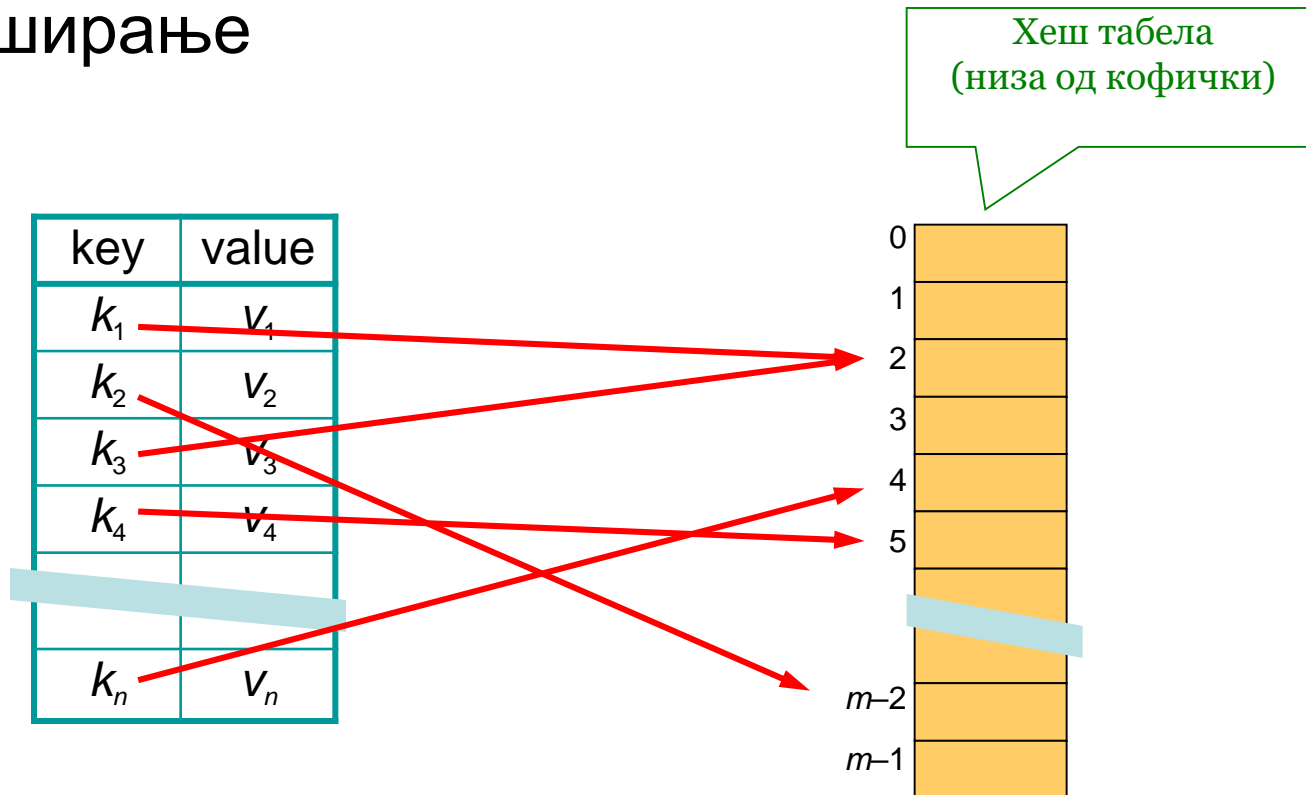
key	value
k_1	v_1
k_2	v_2
k_3	v_3
k_4	v_4
k_n	v_n

Хеш табела
(низ од кофички)



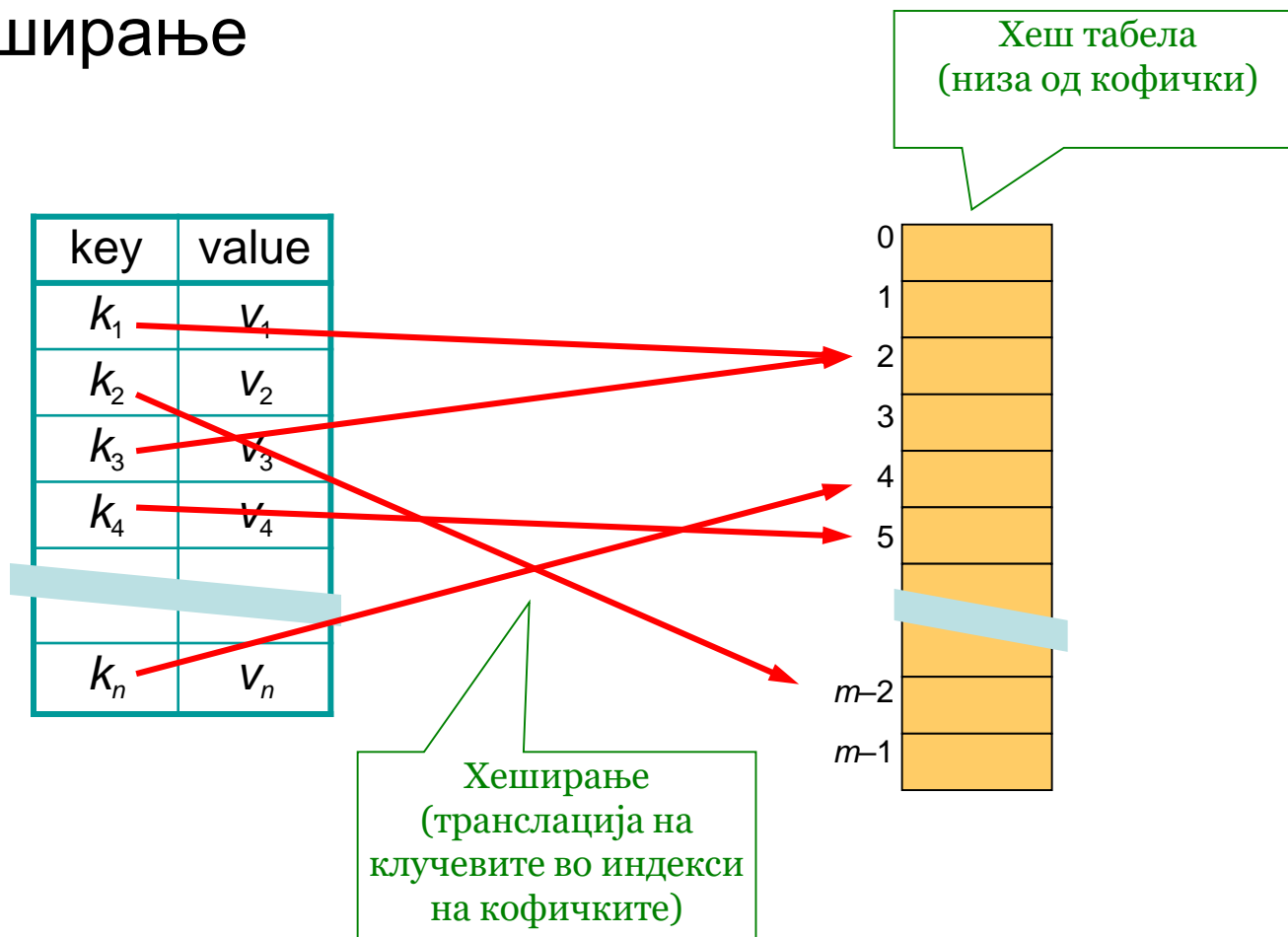
Илустрација

Хеширање



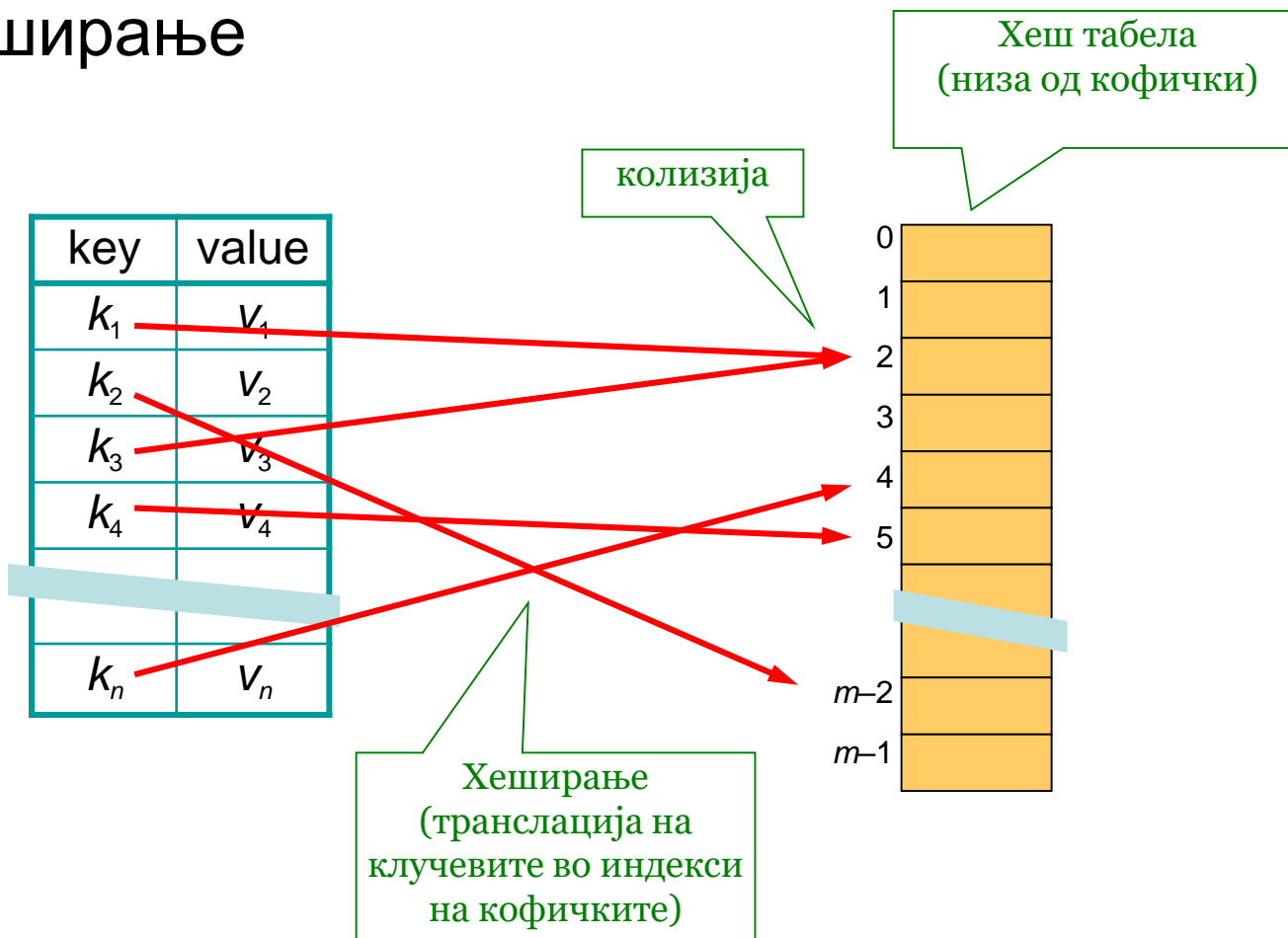
Илустрација

Хеширање



Илустрација

Хеширање



Принципи на хеширање

- ❑ Секој клуч k има своја кофичка, односно онаа кофичка со индекс $hash(k)$. Се нарекува уште и домашна кофичка
- ❑ Нов елемент со клуч k се додава во домашната кофичка на клучот k .
- ❑ За да се најде елемент со клуч k , потребно е да се пребара во домашната кофичка за тој клуч
- ❑ За да се избрише елемент со клуч k , исто така се бара во домашната кофичка на тој клуч

Принципи на хеширање

- ❑ Хеш функцијата треба да биде конзистентна :
 $k_1 = k_2$ следи $hash(k_1) = hash(k_2)$.
 - ❑ Во општ случај, оваа функција прави пресликување многу-спрема-еден
 - ❑ Тоа значи дека за различни клучеви може да се добие иста домашна кофичка :
 $k_1 \neq k_2$ но $hash(k_1) = hash(k_2)$.
- Ова се нарекува **колизија**.

Принципи на хеширање

- ❑ Хеш функцијата треба да биде конзистентна :
 $k_1 = k_2$ следи $hash(k_1) = hash(k_2)$.
 - ❑ Во општ случај, оваа функција прави пресликување многу-спрема-еден
 - ❑ Тоа значи дека за различни клучеви може да се добие иста домашна кофичка :
 $k_1 \neq k_2$ но $hash(k_1) = hash(k_2)$.
- Ова се нарекува **колизија**.

Секогаш треба да е тежнее кон хеш функции кои ретко доведуваат до колизии!

Пример: Хеширање на зборови

- Да претпоставиме дека клучевите се зборови од англискиот јазик.

- Една можна хеш функција:

$$m = 26$$

$$\text{hash}(w) = (\text{првата буква од } w) - 'A'$$

- Сите зборови кои почнуваат со 'A' ја делат кофичката 0;

...

сите зборови кои почнуваат со 'Z' ја делат кофичката 25.

Пример: Хеширање на зборови

- Да претпоставиме дека клучевите се зборови од англискиот јазик.
- Една можна хеш функција:

$$m = 26$$

$$\text{hash}(w) = (\text{првата буква од } w) - 'A'$$

- Сите зборови кои почнуваат со 'A' ја делат кофичката 0;

...

сите зборови кои почнуваат со 'Z' ја делат кофичката 25.

Ова е добро за илустрација, но е многу лошо за практична примена: во некои кофички колизиите ќе бидат многу чести.

Начини на имплементација

- ❑ **Хеш табела со затворени кофички (Closed-bucket hash table):**
 - Во секоја кофичка може да се сместат по повеќе елементи.
 - Кофичките се потполно меѓусебно одвоени.
- ❑ **Хеш табела со отворени кофички (Open-bucket hash table):**
 - Секоја кофичка може да содржи најмногу еден елемент.
 - Ако се појави колизија, новиот елемент се преместува во друга кофичка.

Хеш табела со затворени кофички

❑ Хеш табела со затворени кофички (Closed-bucket hash table):

- Во секоја кофичка може да се сместат по повеќе елементи.
- Кофичките се потполно мреѓусебно одвоени. Нема претекување, затоа се викаат затворени

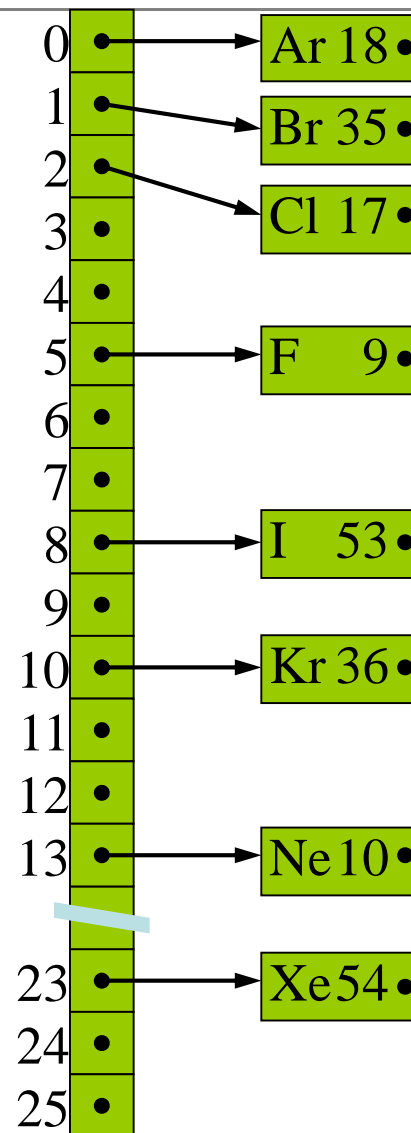
❑ Наједноставна имплементација:

- секоја кофичка е единечно поврзана листа,
- имаме низа од единечно поврзани листи

Хеш табела со затворени кофички

Илустрација (без колизии)

element	number
F	9
Ne	10
Cl	17
Ar	18
Br	35
Kr	36
I	53
Xe	54



Хеширање на хемиски елементи:

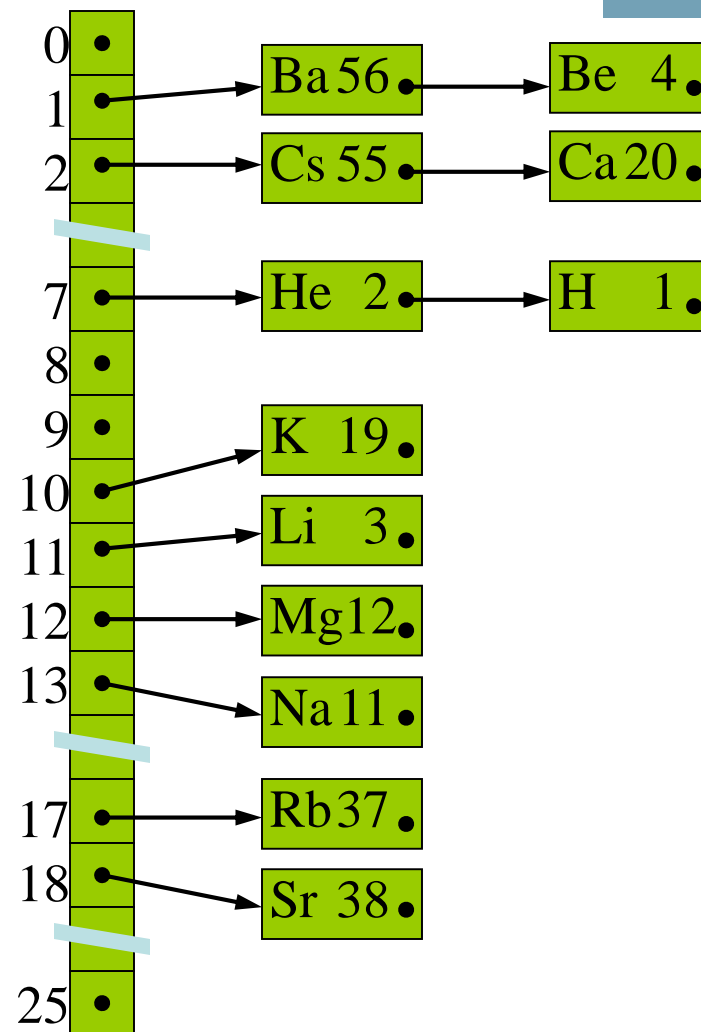
$m = 26$

$hash(e) = (\text{првата буква од } e) - 'A'$

Хеш табела со затворени кофички

□ Илустрација (со колизии):

element	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56



Пребарување во СВНТ

- Треба да се пронајде елементот со клуч *target-key*
 1. $b \leftarrow \text{hash}(\text{target-key})$
 2. Во домашната кофичката b барај елемент чиј клуч е еднаков со *target-key*, ако таков постои

Вметнување во СВНТ

- За да се вметне елементот (key , val) во СВНТ:
 1. $b \leftarrow \text{hash}(key)$
 2. Вметни го елементот (key , val) во листата на кофичката за b , заменувајќи ја со val претходната вредност за клучот key , ако таков претходно постоел
 3. Крај.

Бришење од СВНТ

- За да се избрише запис со клуч `key` од СВНТ:
 1. $b \leftarrow \text{hash}(\text{key})$
 2. Од листата на кофичката за b избриши го елементот со клуч еднаков на `key` ако таков постои
 3. Крај.

Анализа на СВНТ

- Броиме споредби:
Нека бројот на елементи е n .
- Во најдобар случај, да претпоставиме дека нема кофичка со повеќе од 2 елементи:
максимум број на споредби = 2
Сложеност во најдобар случај $O(1)$.
- Во најлош случај, сите елементи се во една кофичка:
максимум број на споредби = n
Сложеност во најлош случај $O(n)$.

Дизајн на СВНТ

- Дизајнот на СВНТ се состои од
 - Избор на бројот на кофички m
 - Избор на хеш функција $hash()$
- Цели
 - Колизииите да бидат ретки
 - Елементите подеднакво да се дистрибуираат помеѓу сите кофички, така што ќе немаме преполнети и полупразни кофички

Избор на бројот на кофички

- ❑ Фактор на пополнетост (**load factor**) на хеш табелата е средниот број на елементи по кофичка, односно n/m .
- ❑ Ако n е познат или барем предвидлив, треба да се избере m така што овој фактор да биде помеѓу 0.5 и 0.75.
 - Помал фактор значи трошење на простор.
 - Голем фактор значи дека ќе има кофички со многу елементи.
- ❑ Избор на m да биде прост број.
 - При хеширањето, често се прави аритметика по модул m .
 - Ако m е прост, дистрибуцијата на елементите низ кофичките ќе биде многу подобра.

Избор на хеш функција

- ❑ Хеш функцијата треба да биде ефикасна (да извршува неколку основи и брзи операции).
- ❑ Хеш функцијата треба да обезбеди рамномерна распределба на елементите по кофичките, независно дали постојат одредени образци на повторување во клучевите.
- ❑ Баланс:
 - Забрзување така што ќе се користи само дел од клучот во пресметката
 - Но ова може да доведе до колиизи заради повторувачки образци во тој дел

Пример: хеш табела на зборови

- ❑ Нека имаме 1000 зборови од англискиот јазик
- ❑ По статистичката обработка на зборовите сме добиле:
 - Различен број на појаување на буквите:
 - А, Е, I, N, S, Т се чести
 - Q, X, Z се ретки
 - Различни должини на зборовите:
 - Зборовите со должина од 4 до 8 букви се чести
 - Зборовите со други должини се ретки

Пример: хеш табела на зборови

- ❑ Нека имаме 1000 зборови од англискиот јазик
- ❑ По статистичката обработка на зборовите сме добиле:
 - Различен број на појаување на буквите:
 - А, Е, И, N, S, Т се чести
 - Q, X, Z се ретки
 - Различни должини на зборовите:
 - Зборовите со должина од 4 до 8 букви се чести
 - Зборовите со други должини се ретки

Како да се изберат најсоодветните вредности за m и за хеш функцијата?

Пример: хеш табела на зборови

- $hash(w)$ може да зависи од почетната буква од зборовите и-или од должината на зборовите
 - На пример $m = 20$, $hash(w) =$ должина на $w - 1$.
 - Премалку кофички. Load factor = $1000/20 = 50$.
 - Многу лоша дистрибуција.
 - На пример $m = 26$, $hash(w) =$ прва буква од $w - 'A'$.
 - Премалку кофички .
 - Многу лоша дистрибуција.

Пример: хеш табела на зборови

- На пример $m = 520$, $hash(w) = 26 \times (\text{должина на } w - 1) + (\text{прва буква од } w - 'A')$.
 - Премалку кофички. Load factor = $1000/520 \approx 1.9$.
 - Лоша дистрибуција. Бидејќи малку зборови имаат должина 0-2, кофичките 0-51 ќе имаат малку елементи. Бидејќи има малку зборови кои почнуваат со Z, кофичките 25, 51, 77, 103, ... ќе имаат малку елементи.
- На пример $m = 1499$, $hash(w) = (\text{тежинска сума на буквите од } w) \text{ modulo } m$
 - пр., $(c_1 \times \text{прва буква } w + c_2 \times \text{втора буква од } w + \dots) \text{ modulo } m$
 - + Добар број на кофички. Load factor ≈ 0.67 .
 - + Релативно добра дистрибуција.

Хеш табела со отворени кофички

❑ Хеш табела со отворени кофички (Open-bucket hash table):

- Секоја кофичка може да содржи најмногу еден елемент.
- Ако се појави колизија, новиот елемент се преместува во друга кофичка.

❑ Секоја кофичка може да се најде во една од трите состојби:

- **никогаш-зафатена** (никогаш не содржела елемент)
- **зафатена** (моментално има елемент)
- **претходно-зафатена** (содржела елемент, кој е избришан и во моментов нема нов).

Хеш табела со отворени кофички

□ Илустрација (без колизии):

element	number
F	9
Ne	10
Cl	17
Ar	18
Br	35
Kr	36
I	53
Xe	54

0	Ar 18	зафатена
1	Br 35	
2	Cl 17	
3		никогаш зафатена
4		
5	F 9	
6		
7		
8	I 53	
9		
10	Kr 36	
11		
12		
13	Ne 10	
14		
22		
23	Xe 54	
24		
25		

Хеш табела со отворени кофички

□ Илустрација (со колизии):

element	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0		
1	Be	4
2	Ca	20
3	Cs	55
4	Ba	56
5		
6		
7	H	1
8	He	2
9		
10	K	19
11	Li	3
12	Mg	12
13	Na	11
...		
17	Rb	37
18	Sr	38
...		
25		

кластер

кластер

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
17	
18	
25	

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0	
1	
2	
3	
4	
5	
6	
7	H 1
8	
9	
10	
11	
12	
13	
17	
18	
25	

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0	
1	
2	
3	
4	
5	
6	
7	H 1
8	He 2
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0	
1	
2	
3	
4	
5	
6	
7	H 1
8	He 2
9	
10	
11	Li 3
12	
13	
17	
18	
25	

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0	
1	Be 4
2	
3	
4	
5	
6	
7	H 1
8	He 2
9	
10	
11	Li 3
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0	
1	Be 4
2	
3	
4	
5	
6	
7	H 1
8	He 2
9	
10	
11	Li 3
12	
13	Na 11
17	
18	
25	

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0	
1	Be 4
2	
3	
4	
5	
6	
7	H 1
8	He 2
9	
10	
11	Li 3
12	Mg 12
13	Na 11
17	
18	
25	

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0	
1	Be 4
2	
3	
4	
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	
18	
25	

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0	
1	Be 4
2	Ca 20
3	
4	
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
/	
17	
18	
/	
25	

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0	
1	Be 4
2	Ca 20
3	
4	
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	
25	

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0	
1	Be 4
2	Ca 20
3	
4	
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	Sr 38
25	

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0	
1	Be 4
2	Ca 20
3	Cs 55
4	
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	Sr 38
25	

Полнење на ОВНТ

- Анимација:

<u>element</u>	number
H	1
He	2
Li	3
Be	4
Na	11
Mg	12
K	19
Ca	20
Rb	37
Sr	38
Cs	55
Ba	56

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	Sr 38
25	

Пребарување во ОВНТ

- За да се најде дали некоја кофичка во ОВНТ е зафатена со елемен чиј клуч е *target-key*:
 1. $b \leftarrow \text{hash}(\text{target-key})$
 2. Повторувај:
 - 2.1. Ако кофичката b е *никогаш-зафатена*:
 - 2.1.1. Прекини го пребарувањето со одговор *НЕМА*.
 - 2.2. Ако кофичката b е зафатена со клуч кој е еднаков со *target-key*:
 - 2.2.1. Прекини го пребарувањето со одговор b .
 - 2.3. Ако кофичката b е *претходно-зафатена*, или е зафатена со клуч кој не е еднаков со *target-key*:
 - 2.3.1. Зголеми го b по modulo m .

Пребарување во ОВНТ

☐ Илустрација:

Пребарување во ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	Sr 38
19	
25	

Пребарување во ОВНТ

□ Илустрација:

Пребарувај Mg:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	Sr 38
19	
25	

Пребарување во ОВНТ

□ Илустрација:

Пребарувај Mg: →

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	Sr 38
19	
25	

Пребарување во ОВНТ

□ Илустрација:

Пребарувај He:

Пребарувај Mg: →


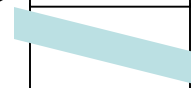
0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	Sr 38
19	
25	

Пребарување во ОВНТ

□ Илустрација:

Пребарувај He: 

Пребарувај Mg: 

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
	
17	Rb 37
18	Sr 38
19	
	
25	


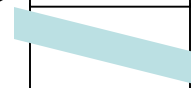
Пребарување во ОВНТ

□ Илустрација:

Пребарувај Ва:

Пребарувај Не: 

Пребарувај Мг: 

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
	
17	Rb 37
18	Sr 38
19	
	
25	

Пребарување во ОВНТ

□ Илустрација:

Пребарувај Ва:



Пребарувај Не:



Пребарувај Mg:




0		
1	Be	4
2	Ca	20
3	Cs	55
4	Ba	56
5		
6		
7	H	1
8	He	2
9		
10	K	19
11	Li	3
12	Mg	12
13	Na	11
17	Rb	37
18	Sr	38
19		
25		

Пребарување во ОВНТ


□ Илустрација:

Пребарувај Ва:




0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	Sr 38
19	
25	


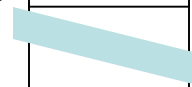
Пребарувај Не:



Пребарувај Mg:




Пребарувај Ра:

Пребарување во ОВНТ


□ Илустрација:

Пребарувај Ва:




0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	Sr 38
19	
25	


Пребарувај Не:



Пребарувај Mg:



Пребарувај Ра:



Вметнување во ОВНТ

- За вметнување на елементот (key, val) во ОВНТ:
 1. $b \leftarrow hash(key)$
 2. Повторувај:
 - 2.1. Ако кофичката b е *никогаш-зафатена*:
 - 2.1.1. Ако кофичката b е последната *никогаш-зафатена* кофичка, тогаш ОВНТ е преполнето.
 - 2.1.2. Нека кофичката b биде зафатена од (key, val) .
 - 2.1.3. Заврши.
 - 2.2. Ако кофичката b е *претходно-зафатена*, или е зафатена од клуч кој е еднаков со key :
 - 2.2.1. Нека кофичката b биде зафатена од (key, val) .
 - 2.2.2. Заврши.
 - 2.3. Ако кофичката b е зафатена од клуч кој е различен од key :
 - 2.3.1. Зголеми го b по modulo m .

Вметнување во ОВНТ

☐ Илустрација:

Вметнување во ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	Sr 38
25	

Вметнување во ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	Sr 38
25	

Вметнување
на (Fr, 87):

Вметнување во ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Вметнување
на (Fr, 87):

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	Fr 87
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Вметнување во ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Вметнување
на (Fr, 87):



0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	Fr 87
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Вметнување во ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Вметнување
на (Fr, 87):



0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	Fr 87
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Вметнување
на (B, 5):

Вметнување во ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Вметнување
на (Fr, 87):



0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	Fr 87
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Вметнување
на (B, 5):

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	Fr 87
6	B 5
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Вметнување во ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Вметнување
на (Fr, 87):



0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	Fr 87
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Вметнување
на (B, 5):



0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	Fr 87
6	B 5
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Бришење од ОВНТ

- ❑ За бришење на елемент (ако постои) чиј клуч е еднаков на *key* од ОВНТ:
 1. $b \leftarrow \text{hash}(\text{key})$.
 2. Повторувај:
 - 2.1. Ако кофичката b е *никогаш-зафатена*:
 - 2.1.1. Прекини.
 - 2.2. Ако кофичката b е зафатена со клуч еднаков со *key*:
 - 2.2.1. Нека b стане *претходно-зафатена*.
 - 2.2.2. Прекини.
 - 2.3. Ако кофичката b е претходно-зафатена, или е зафатена со клуч кој е различен од *key*:
 - 2.3.1. Зголеми го b по modulo m .

Бришење од ОВНТ

☐ Илустрација:

Бришење од ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	Sr 38
25	

Бришење од ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
17	Rb 37
18	Sr 38
25	

Бришење на
Ca:

Бришење од ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Бришење на
Ca:

0	
1	Be 4
2	
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Бришење од ОВНТ

□ Илустрација:

0		0	
1	Be 4	1	Be 4
2	Ca 20	2	
3	Cs 55	3	Cs 55
4	Ba 56	4	Ba 56
5		5	
6		6	
7	H 1	7	H 1
8	He 2	8	He 2
9		9	
10	K 19	10	K 19
11	Li 3	11	Li 3
12	Mg 12	12	Mg 12
13	Na 11	13	Na 11
17	Rb 37	17	Rb 37
18	Sr 38	18	Sr 38
25		25	

Бришење на
Ca:



Бришење од ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
/	
17	Rb 37
18	Sr 38
/	
25	

Бришење на
Ca:

0	
1	Be 4
2	
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
/	
17	Rb 37
18	Sr 38
/	
25	

претходно-
зафатена

Бришење од ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Бришење на
Ca:



0	
1	Be 4
2	
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
...	
17	Rb 37
18	Sr 38
...	
25	

Бришење на
Ba:

претходно-
зафатена

Бришење од ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
/	
17	Rb 37
18	Sr 38
/	
25	

Бришење на
Ca:



0	
1	Be 4
2	
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
/	
17	Rb 37
18	Sr 38
/	
25	

Бришење на
Ba:

претходно-
зафатена

0	
1	Be 4
2	
3	Cs 55
4	
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
/	
17	Rb 37
18	Sr 38
/	
25	

Бришење од ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
/	
17	Rb 37
18	Sr 38
/	
25	

Бришење на
Ca:

0	
1	Be 4
2	
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
/	
17	Rb 37
18	Sr 38
/	
25	

Бришење на
Ba:

претходно-
зафатена

0	
1	Be 4
2	
3	Cs 55
4	
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
/	
17	Rb 37
18	Sr 38
/	
25	

Бришење од ОВНТ

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
/	
17	Rb 37
18	Sr 38
/	
25	

Бришење на
Ca:

0	
1	Be 4
2	
3	Cs 55
4	Ba 56
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
/	
17	Rb 37
18	Sr 38
/	
25	

Бришење на
Ba:

0	
1	Be 4
2	
3	Cs 55
4	
5	
6	
7	H 1
8	He 2
9	
10	K 19
11	Li 3
12	Mg 12
13	Na 11
/	
17	Rb 37
18	Sr 38
/	
25	

претходно-
зафатена

претходно-
зафатена

Анализа на ОВНТ

- ❑ Анализа на алгоритмите за пребарување/вметнување/бришење (броејќи споредби):
Нека бројот на елементи е n .
- ❑ Во најдобар случај нека не постои кластер со повеќе од 4 соседни пополнети кофички:
Максимален број на споредби = 4
Сложеност $O(1)$.
- ❑ Во најлош случај, нека има еден кластер со сите n елементи:
Максимален број на споредби = n
Сложеност $O(n)$.

Дизајн на ОВНТ

□ Дизајнот на ОВНТ опфаќа:

- Избор на број на кофички m
- Избор на хеш функција $hash$
- Избор на должина на чекор s (ќе го објасниме подоцна).

□ Цели:

- Колизииите треба да бидат ретки
- Елементите треба да бидат добро распределени низ табелата, така што нема да има кластери со повеќе од 4 елементи.

Избор на број на кофички

- ❑ Да се потсетиме: Фактор на пополнетост (**load factor**) на хеш табелата е средниот број на елементи по кофичка, односно n/m
- ❑ Ако n е познат или барем предвидлив, треба да се избере m така што овој фактор да биде помеѓу 0.5 и 0.75.
 - Помал фактор значи трошење на простор.
 - Голем фактор значи дека ќе има кофички со многу елементи.
- ❑ Избор на m да биде прост број.

Избор на хеш функција

- ❑ Треба да биде ефикасна.
- ❑ Треба да ги дистрибуира елементите подеднакво низ сите кофички, со што е можно помалку долги кластери.
 - Во ОВНТ со должина на чекор $s = 1$, кластер ќе се формира ако неколку елементи се пресликаат во исти или соседни кофички.

Избор на должина на чекорот

- За избегнување на колизија, алгоритмот за вметнување (а потоа и пребарување и бришење), го зголемуваат индексот на кофичката за 1 и се обидуваат повторно.
- Овој фактор се нарекува должината на чекорот, s , што претставува број за кој се зголемува индексот на следната разгледана кофичка.

Избор на должина на чекорот

- До сега претпоставувавме дека е $s = 1$.
- Алтернативно, можеме да користиме фиксно $s > 1$.
- Изберете го m да биде прост, а s да биде во рангот $2 \dots m-1$.
 - Во тој случај, s и m немаат заеднички фактори.
 - Ако имаат, на пример $m = 10$ и $s = 2$, вообичаена патека за пребарување и сместување ќе биде 6–8–0–2–4, при што другите кофички ќе останат празни!

Двојно хеширање

- ❑ Уште подобро, нека за различните клучеви има различни должини на чекорот. (Секако, секој клуч секогаш ќе има иста вредност на чекорот)
- ❑ **Двојно хеширање:** За вметнување/пребарување/бришење на елемент со клуч k , се пресметува s од k , користејќи втора хеш функција $s = \text{step}(k)$.
- ❑ Во примерот со хемиските елементи:

$$m = 23$$

$$\text{hash}(e) = (\text{првата буква на } e - 'A') \bmod m$$



$$\begin{aligned} \text{step}(e) &= 1 \text{ ако името е со една буква} \\ &2 + (\text{втората буква од } e - 'a') \bmod 21 \end{aligned}$$

Двојно хеширање

□ Илустрација:

Двојно хеширање

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
	
17	Rb 37
18	Sr 38
	
22	

Двојно хеширање

❑ Илустрација:

0	
1	Be 4
2	Ca 20
3	
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
17	Rb 37
18	Sr 38
22	

Вметнување
на (Ba, 56):

Двојно хеширање

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
...	
17	Rb 37
18	Sr 38
...	
22	

Вметнување
на (Ba, 56):

$hash(Ba) = 1$
 $step(Ba) = 2$

Двојно хеширање

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
...	
17	Rb 37
18	Sr 38
...	
22	

Вметнување
на (Ba, 56):

$hash(Ba) = 1$
 $step(Ba) = 2$

0	
1	Be 4
2	Ca 20
3	Ba 56
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
...	
17	Rb 37
18	Sr 38
...	
22	

Двојно хеширање

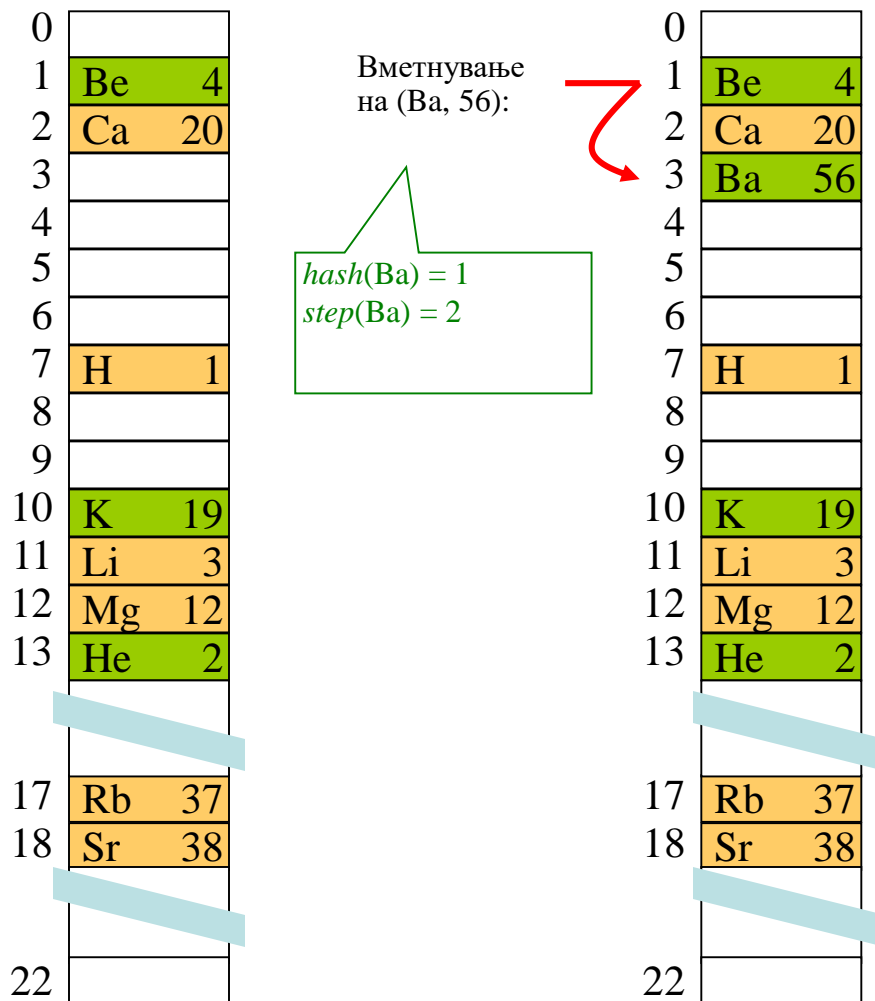
□ Илустрација:

0	
1	Be 4
2	Ca 20
3	
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
17	Rb 37
18	Sr 38
22	

Вметнување
на (Ba, 56):

$hash(Ba) = 1$
 $step(Ba) = 2$

0	
1	Be 4
2	Ca 20
3	Ba 56
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
17	Rb 37
18	Sr 38
22	



Двојно хеширање

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
...	
17	Rb 37
18	Sr 38
...	
22	

Вметнување
на (Ba, 56):

$hash(Ba) = 1$
 $step(Ba) = 2$



0	
1	Be 4
2	Ca 20
3	Ba 56
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
...	
17	Rb 37
18	Sr 38
...	
22	

Вметнување
на (Cs, 55):

Двојно хеширање

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
...	
17	Rb 37
18	Sr 38
...	
22	

Вметнување
на (Ba, 56):

$hash(Ba) = 1$
 $step(Ba) = 2$



0	
1	Be 4
2	Ca 20
3	Ba 56
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
...	
17	Rb 37
18	Sr 38
...	
22	

Вметнување
на (Cs, 55):

$hash(Cs) = 2$
 $step(Cs) = 20$

Двојно хеширање

Илустрација:

0	
1	Be 4
2	Ca 20
3	
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
...	
17	Rb 37
18	Sr 38
...	
22	

Вметнување
на (Ba, 56):

$hash(Ba) = 1$
 $step(Ba) = 2$



0	
1	Be 4
2	Ca 20
3	Ba 56
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
...	
17	Rb 37
18	Sr 38
...	
22	

Вметнување
на (Cs, 55):

$hash(Cs) = 2$
 $step(Cs) = 20$

0	
1	Be 4
2	Ca 20
3	Ba 56
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
...	
17	Rb 37
18	Sr 38
...	
22	Cs 55

Двојно хеширање

□ Илустрација:

0	
1	Be 4
2	Ca 20
3	
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
...	
17	Rb 37
18	Sr 38
...	
22	

Вметнување
на (Ba, 56):

$hash(Ba) = 1$
 $step(Ba) = 2$

0	
1	Be 4
2	Ca 20
3	Ba 56
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
...	
17	Rb 37
18	Sr 38
...	
22	

Вметнување
на (Cs, 55):

$hash(Cs) = 2$
 $step(Cs) = 20$

0	
1	Be 4
2	Ca 20
3	Ba 56
4	
5	
6	
7	H 1
8	
9	
10	K 19
11	Li 3
12	Mg 12
13	He 2
...	
17	Rb 37
18	Sr 38
...	
22	Cs 55

Вметнување со двојно хеширање

□ За да се вметне елементот со (key, val) во ОВНТ:

1. $b \leftarrow hash(key)$, и $s \leftarrow step(key)$.
2. Повторувај:
 - 2.1. Ако кофичката b е *никогаш-зафатена*:
...
 - 2.2. Ако кофичката b е *претходно-зафатена*, или е зафатена со клуч еднаков со key :
...
 - 2.3. Ако кофичката b е зафатена со клуч чија вредност е различна од key :
 - 2.3.1. Зголеми го b преку s , по modulo m .

Хеш табели во пракса

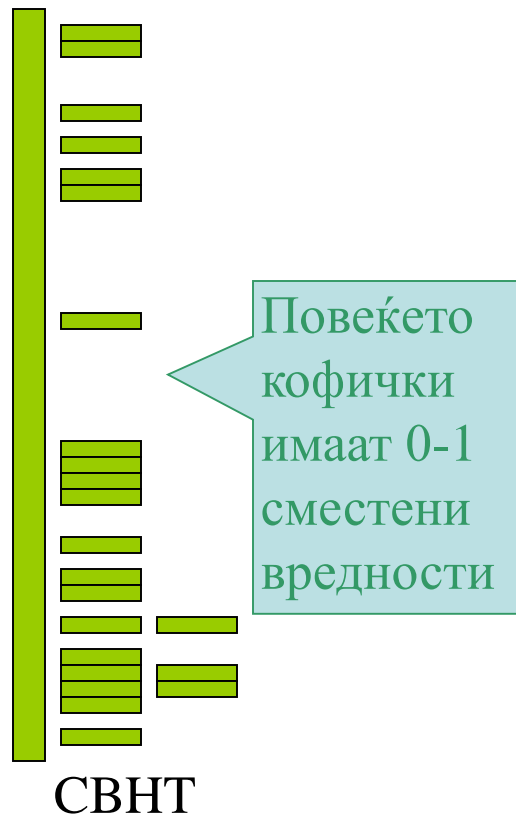
- ❑ Следните тестови ги споредуваат перформансите на три хеш табели, секоја со m кофички:
 - СВНТ
 - ОВНТ со чекор $s = 1$
 - ОВНТ со чекор s добиен со двојно хеширање.
- ❑ Хеш функцијата е избрана да ги распределува клучевите униформно низ m кофички.
- ❑ Во секој тест, сите три табели се полнат со истото множество од случајни броеви.

Хеш табели во пракса

□ Тест со $m = 47$ и $n = 23$ (load factor ≈ 0.5):

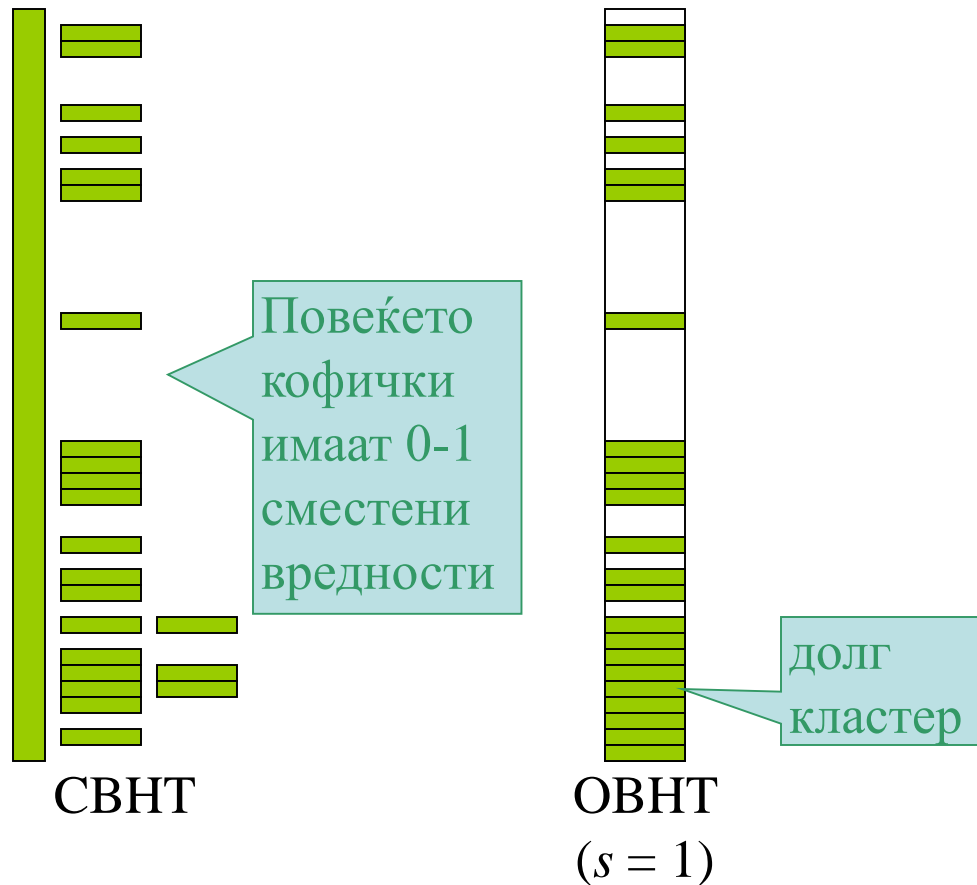
Хеш табели во пракса

□ Тест со $m = 47$ и $n = 23$ (load factor ≈ 0.5):



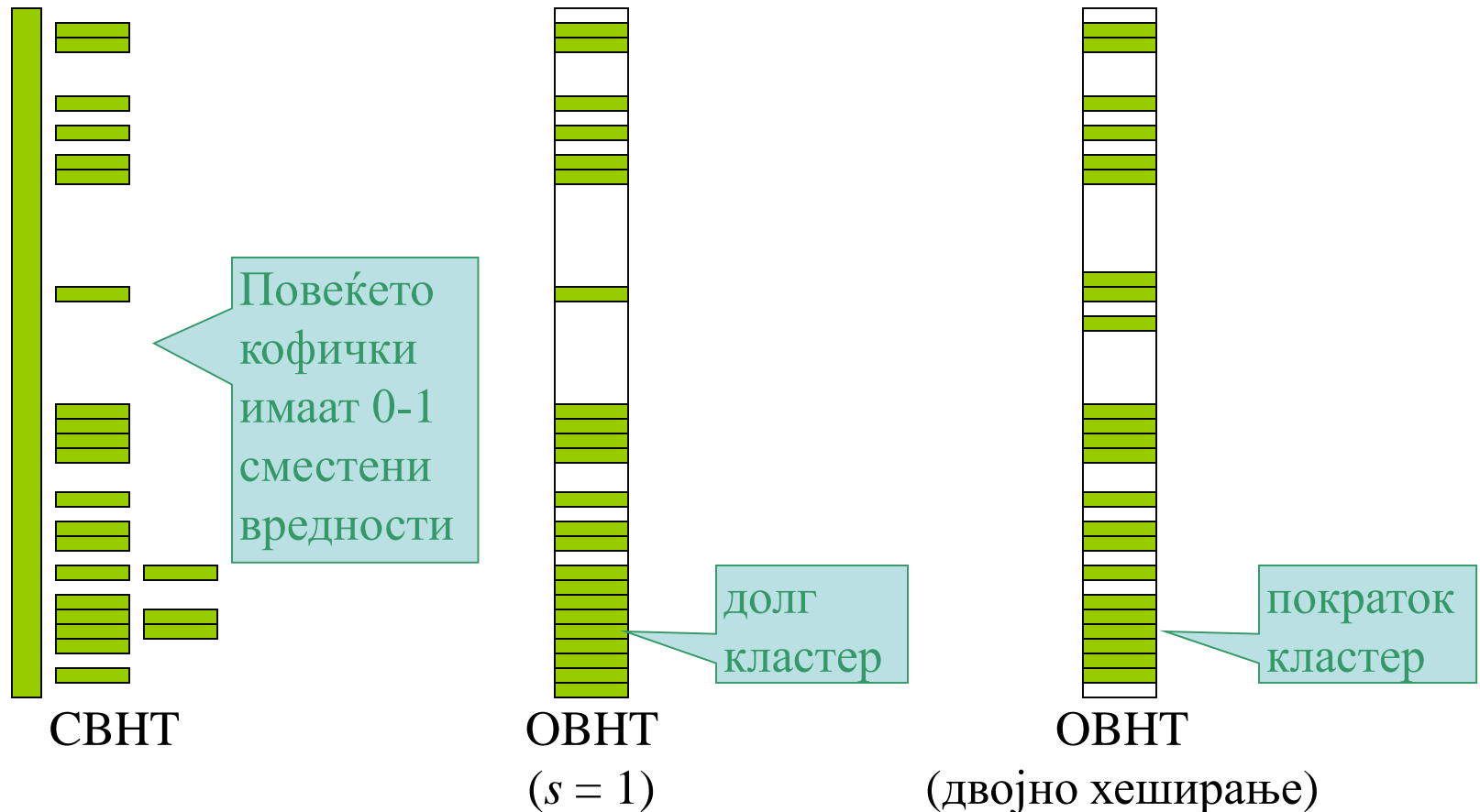
Хеш табели во пракса

□ Тест со $m = 47$ и $n = 23$ (load factor ≈ 0.5):



Хеш табели во пракса

□ Тест со $m = 47$ и $n = 23$ (load factor ≈ 0.5):

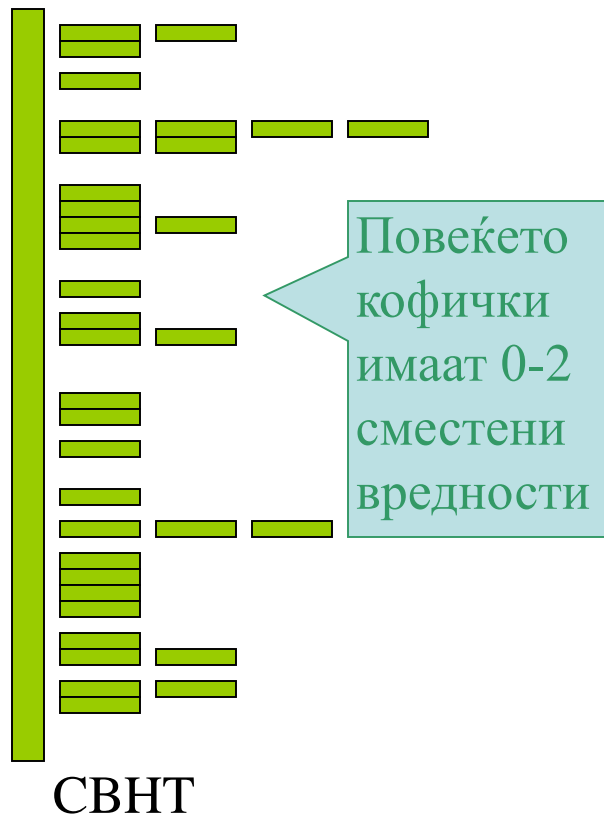


Хеш табели во пракса

□ Тест со $m = 47$ и $n = 36$ (load factor ≈ 0.75):

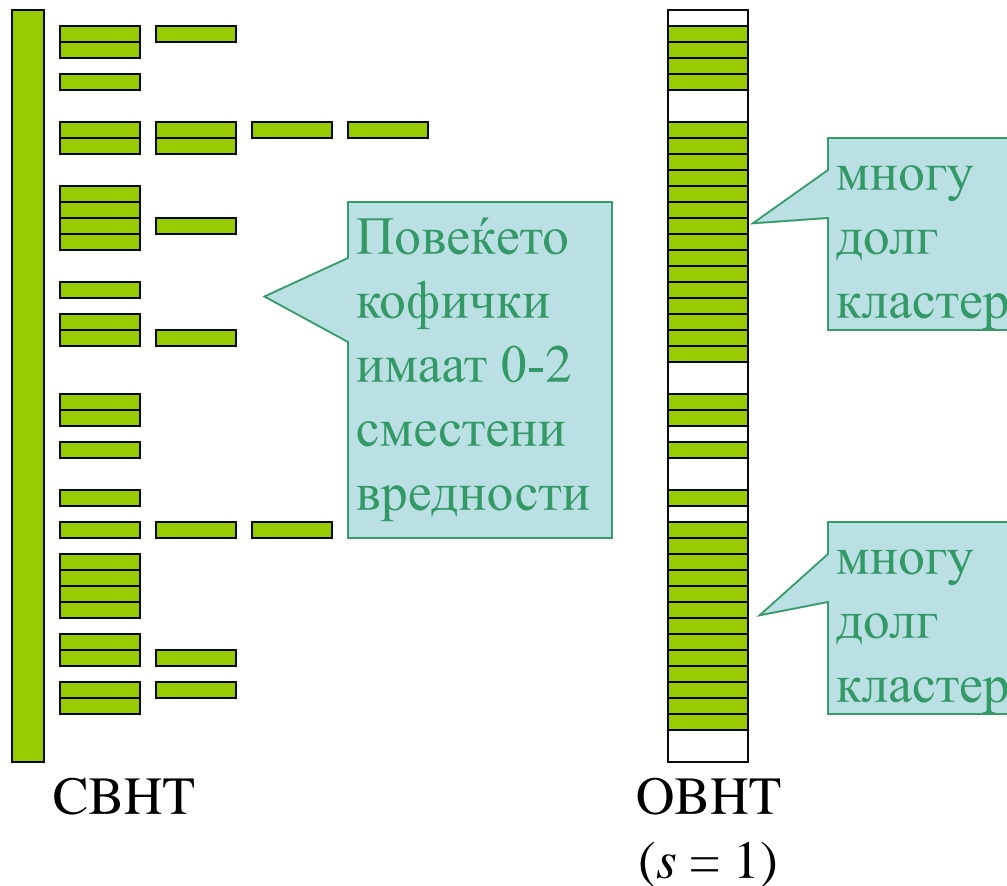
Хеш табели во пракса

□ Тест со $m = 47$ и $n = 36$ (load factor ≈ 0.75):



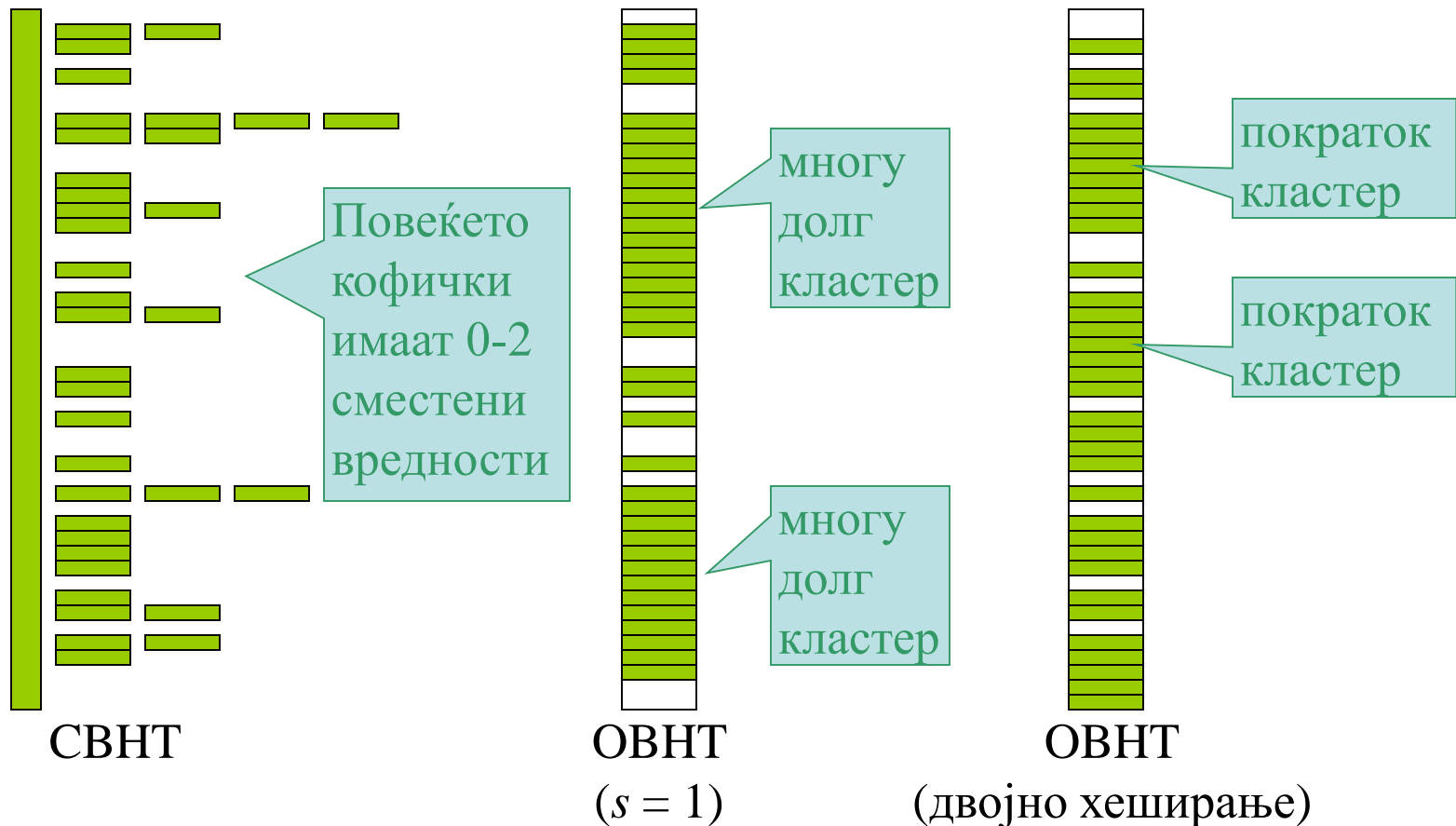
Хеш табели во пракса

□ Тест со $m = 47$ и $n = 36$ (load factor ≈ 0.75):



Хеш табели во пракса

□ Тест со $m = 47$ и $n = 36$ (load factor ≈ 0.75):



Пример: студентски досиеа

- ❑ Да разгледаме пример на некој универзитет и неговите студентски досиеа.
- ❑ Секоја година околу 1500 нови студенти се запишуваат на овој универзитет, а студираат во просек 4 години.
- ❑ Секој студент има свој единствен id, во форма *yydddd* (каде *yy* се последните две цифри од годината на запишување, а *dddd* е сериски број).
- ❑ Записите за студентите се чуваат во хеш табела.

Пример: студентски досиеа

- ❑ Нека $m = 100$, $hash(id) =$ првите 2 цифри од id .
 - Многу малку кофички. Load factor $\approx 6000/100 \approx 60$.
 - Многу лоша дистрибуција Во учебната 2012–13, сите id -а почнуваат со 09, 10, 11, or 12.
- ❑ Нека $m = 10000$, $hash(id) =$ последните 4 цифри од id .
 - + Добар број на кофички. Load factor $\approx 6000/10000 \approx 0.6$.
 - Лоша дистрибуција. Скоро сите id -а завршуваат со 0000...1500.
- ❑ Нека $m = 9997$, $hash(id) = id \bmod m$.
 - + Добар број на кофички. Load factor $\approx 6000/9997 \approx 0.6$.
 - + Добра дистрибуција (бидејќи m е прост).

Пример: студентски досиеа

- ❑ Нека имаме ОВНТ со $s = 1$.
 - Четири кластери со по 1500 записи.
- ❑ Нека имаме ОВНТ со $s = 2000$.
 - + Избегнува кластери.
- ❑ Нека имаме ОВНТ со двојно хеширање.
 - + Избегнува кластери.

Имплементација на множества со СВНТ

□ Сложеност:

Операција	Алгоритам	Временска комплексност
пребарување	СВНТ search	$O(1)$ најдобро $O(n)$ најлошо
вметнување	СВНТ insertion	$O(1)$ најдобро $O(n)$ најлошо
бришење	СВНТ deletion	$O(1)$ најдобро $O(n)$ најлошо