

Aufgabe 4.1:

Lineares Hashing

 $n = 5$, $b = 2$, $bf_s = 0.75$, $h_0(k) = k \bmod 5$, $h_1(k) = k \bmod 10$, $p = 0$

Einfügen : 200, 325, 405, 389, 188, 101, 500, 57

p				
0	1	2	3	4
200 325	101	57	188	389

405 500

 $bf = 8/10 = 0.8$

dann muss man in diesem Fall die Tabelle aufspalten

	p				
0	1	2	3	4	5
200 500	101	57	188	389	325 405

 $bf = 8/12 = 0.66$ $n = 5$, $b = 2$, $bf_s = 0.75$, $h_0(k) = k \bmod 5$, $h_1(k) = k \bmod 10$, $p = 1$

Wir fügen weitere Elemente hinzu: 82, 120

	p				
0	1	2	3	4	5
200 500	101	57 82	188	389	325 405

120

 $bf = 10/12 = 0.83$

hier muss man wieder die Tabelle aufspalten

		p				
0	1	2	3	4	5	6
200 500	101	57 82	188	389	325 405	

120

 $bf = 10/14 = 0.71$ $n = 5$, $b = 2$, $bf_s = 0.75$, $h_0(k) = k \bmod 5$, $h_1(k) = k \bmod 10$, $p = 2$

Wir fügen weitere Elemente hinzu: 133

		p				
0	1	2	3	4	5	6
200 500	101	57 82	188 133	389	325 405	

120

$bf = 11/14 = 0.78$

hier muss man wieder die Tabelle aufspalten:

			p				
0	1	2	3	4	5	6	7
200 500	101	82	188 133	389	325 405		57

120

$bf = 11/16 = 0.68$

$n = 5, b = 2, bf_s = 0.75, h_0(k) = k \bmod 5, h_1(k) = k \bmod 10, p = 3$

Wir fügen weitere Elemente hinzu: 48, 436

			p				
0	1	2	3	4	5	6	7
200 500	101	82	188 133	389	325 405	436	57

120

48

$bf = 13/16 = 0.81$

hier muss man wieder die Tabelle aufspalten:

				p				
0	1	2	3	4	5	6	7	8
200 500	101	82	133	389	325 405	436	57	188 48

120

$bf = 13/18 = 0.72$

$n = 5, b = 2, bf_s = 0.75, h_0(k) = k \bmod 5, h_1(k) = k \bmod 10, p = 4$

Wir fügen weitere Elemente hinzu: 461

				p				
0	1	2	3	4	5	6	7	8
200 500	101 461	82	133	389	325 405	436	57	188 48

120

$bf = 14/18 = 0.77$

hier muss man wieder die Tabelle aufspalten. Nun müssen wir unsere p auf 0, weil die Größe von der Hashtabelle schon 10 ist:

p									
0	1	2	3	4	5	6	7	8	9
200 500	101 461	82	133		325 405	436	57	188 48	389

120

$bf = 14/20 = 0.7$

$n = 5, b = 2, bf_s = 0.75, h_1(k) = k \bmod 10, h_2(k) = k \bmod 20, p = 0$

Wir fügen weitere Elemente hinzu: 364, 223

p									
0	1	2	3	4	5	6	7	8	9
200 500	101 461	82	133 223	364	325 405	436	57	188 48	389

120

$bf = 0.8$

hier muss man wieder die Tabelle aufspalten

	p									
0	1	2	3	4	5	6	7	8	9	10
200 500	101 461	82	133 223	364	325 405	436	57	188 48	389	

120

$$bf = 16/22 = 0.72$$

$$n = 5, b = 2, bf_s = 0.75, h_1(k) = k \bmod 10, h_2(k) = k \bmod 20, p = 1$$

Wir fügen weitere Elemente hinzu: 228

	p									
0	1	2	3	4	5	6	7	8	9	10
200	101	82	133	364	325	436	57	188	389	
500	461		223		405			48		

120

228

$$bf = 17/22 = 0.77$$

Tabelle aufspalten

		p									
0	1	2	3	4	5	6	7	8	9	10	11
200	101	82	133	364	325	436	57	188	389		
500	461		223		405			48			

120

228

$$bf = 17/24 = 0.708$$

$$n = 5, b = 2, bf_s = 0.75, h_1(k) = k \bmod 10, h_2(k) = k \bmod 20, p = 2$$

Wir fügen weitere Elemente hinzu: 199, 123

		p									
0	1	2	3	4	5	6	7	8	9	10	11
200	101	82	133	364	325	436	57	188	389		
500	461		223		405			48	199		

120

123

228

$$bf = 19/24 = 0.79$$

Tabelle aufspalten:

			p									
0	1	2	3	4	5	6	7	8	9	10	11	12
200	101	82	133	364	325	436	57	188	389			
500	461		223		405			48	199			

120

123

228

$$bf = 19/26 = 0.73$$

$$n = 5, b = 2, bf_s = 0.75, h_1(k) = k \bmod 10, h_2(k) = k \bmod 20, p = 3$$

Wir fügen weitere Elemente hinzu: 42

			p									
0	1	2	3	4	5	6	7	8	9	10	11	12
200 500	101 461	82 42	133 223	364	325 405	436	57	188 48	389 199			

120	123	228
-----	-----	-----

$bf = 20/26 = 0.76$

Tabelle aufspalten:

				p									
0	1	2	3	4	5	6	7	8	9	10	11	12	13
200 500	101 461	82 42	133 223	364	325 405	436	57	188 48	389 199				

120	123	228
-----	-----	-----

$bf = 19/26 = 0.73$

$n = 5, b = 2, bf_s = 0.75, h_1(k) = k \bmod 10, h_2(k) = k \bmod 20, p = 4$

				p									
0	1	2	3	4	5	6	7	8	9	10	11	12	13
200 500	101 461	82 42	223 123	364	325 405	436	57	188 48	389 199				133

120	228
-----	-----

Wir fügen weitere Elemente hinzu: 23

				p									
0	1	2	3	4	5	6	7	8	9	10	11	12	13
200 500	101 461	82 42	223 123	364	325 405	436	57	188 48	389 199				133

120	23	228
-----	----	-----

Aufgabe 4.2

a.

$$\begin{aligned} &< 5, 4 > \\ &0 \leq x \leq 19 \\ &x = 5y + z \quad 0 \leq y \leq 3, 0 \leq z \leq 4 \end{aligned}$$

X	y				z				
R	y ₀	y ₁	y ₂	y ₃	z ₀	z ₁	z ₂	z ₃	z ₄
10	0	0	1	0	1	0	0	0	0
5	0	1	0	0	0	0	0	0	0
6	0	1	0	0	0	1	0	0	0
8	0	1	0	0	0	0	0	1	0
7	0	1	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0	0
3	0	0	0	0	0	0	0	1	0
5	0	1	0	0	0	0	0	0	0
6	0	1	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0	0	1

b.

n*m muss alle mögliche Ausprägungen enthalten und n+m muss Bitmap-Vektoren. Damit jede Wert nur einmal durch eine Definition abgebildet werden können und alle Werte abbildbar sein, soll m=n-1 ausgewählt werden.

c.

R	0	1	2	3	4	5	6	7	8	9	10
10	0	0	0	0	0	0	0	0	0	0	1
5	0	0	0	0	0	1	1	1	1	1	1
6	0	0	0	0	0	0	1	1	1	1	1
8	0	0	0	0	0	0	0	0	1	1	1
7	0	0	0	0	0	0	0	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1
3	0	0	0	1	1	1	1	1	1	1	1
5	0	0	0	0	0	1	1	1	1	1	1
6	0	0	0	0	0	0	1	1	1	1	1
4	0	0	0	0	1	1	1	1	1	1	1

d.

beim bereichskodierten Bitmap-Index sind zwei Zugriffe für Punktabfragen nötig, da man gesuchten Wert durch kleiner gleich der allen Werte bekommen kann.