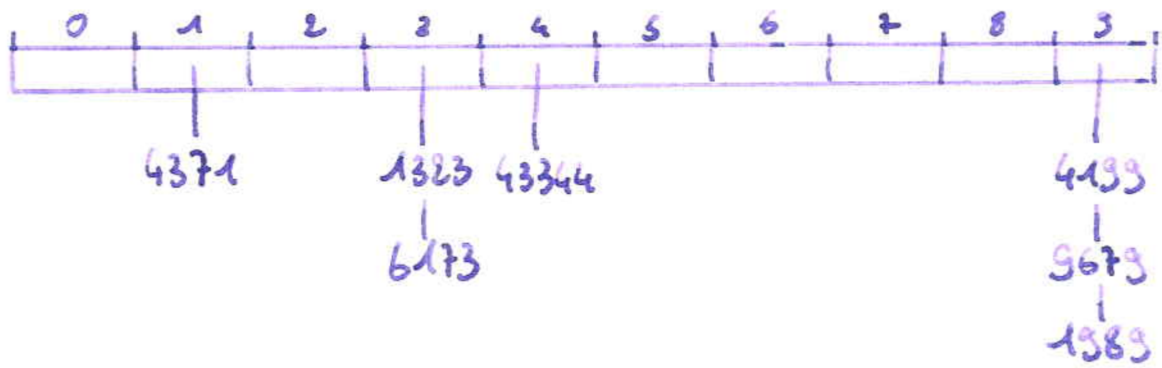
Hoofdstuk 7: Hashing

oef. 1 * external chaining

k	$h(k) = k \bmod 10$
4371	1
1323	3
6173	3
4199	9
43344	4
9679	9
1989	9



* linear probing

0	1	2	3	4	5	6	7	8	9
9679	4371	1989	1323	6173	43344				4199

6173 geeft collisien op pos. 3 → probeer pos. 4 ✓

43344 " " " " 4 → " " 5 ✓

9679 " " " " 9 → " " 0 ✓

1989 " " " " 9 → " " 0 ✗

1989 " " " " 0 → " " 1 ✗

1989 " " " " 1 → " " 2 ✓

* double rehashing

0	1	2	3	4	5	6	7	8	9
	4371		1323	6173	9679		1989	43344	4199

k	$h(k) = k \bmod 10$	$h_2(k) = 7-(k \bmod 7)$
4371	1	
1323	3	
6173	3	$7-(6173 \bmod 7) = 7-6 = 1$
4199	9	
43344	4	$7-(43344 \bmod 7) = 7-0 = 7$
9679	9	$7-(9679 \bmod 7) = 7-5 = 2$
1989	9	$7-(1989 \bmod 7) = 7-1 = 6$

6173 geeft collision op pos. 3 → proberen pos. $\underbrace{h(k)+1.h_2(k)}_{4}$ ✓

43344 " " " " 4 → " " $\underbrace{h(k)+1.h_2(k)}_{1}$ ✗

43344 " " " " 1 → " " $\underbrace{h(k)+2.h_2(k)}_{8}$ ✓

9679 " " " " 9 → " " $\underbrace{h(k)+1.h_2(k)}_{1}$ ✗

9679 " " " " 1 → " " $\underbrace{h(k)+2.h_2(k)}_{3}$ ✗

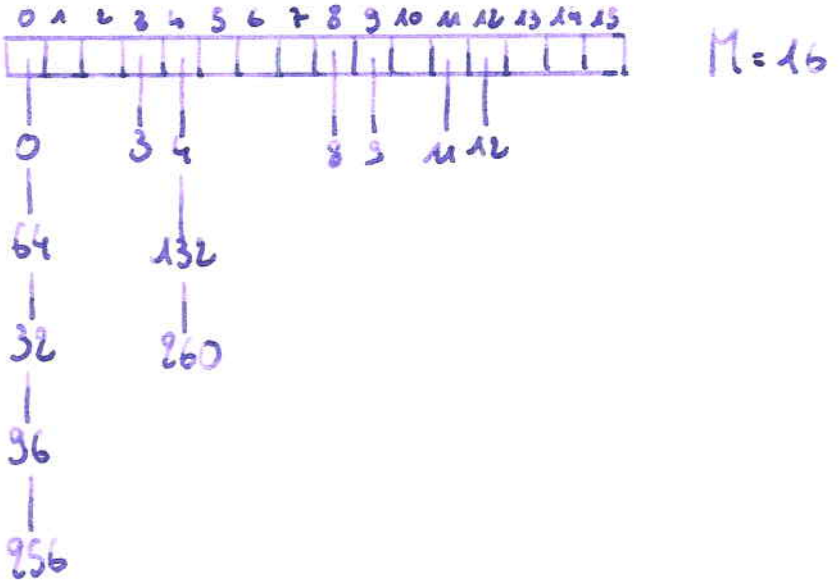
9679 " " " " 3 → " " $\underbrace{h(k)+3.h_2(k)}_{5}$ ✓

1989 " " " " 9 → " " $\underbrace{h(k)+1.h_2(k)}_{5}$ ✗

1989 " " " " 5 → " " $\underbrace{h(k)+2.h_2(k)}_{1}$ ✗

1989 geeft collisie op pos. 1 → proben pos. $\underbrace{h(k)+3.h_2(k)}_{7 \checkmark}$

oef. 2



Fummeling: deel v.d. keys komt op een deel v.d. posities terecht. Hier: positie 0 en 4.
Het probleem doet zich voor wanneer $h(k)$ en M een factor y gemeen hebben. Mogelijke oplossing: kies M priemgetal.

oef. 3 Performantie v. zoeken in een hashtabel met external ch

$$(\underbrace{1}_{\text{iste bucket ... en d.m.v. } h(k)} + \underbrace{?}_{\text{het juiste element zoeken binnen de bucket}})$$

* plain Scheme list: $O(1+\alpha)$
* vectorial sorted list: $O(1+\underbrace{\log(\alpha)}_{\text{omwille van binary search!}})$
* BST: $O(1+\log(\alpha))$ in het beste geval (perfect gebal.)
  $O(1+\alpha)$ in het slechtste geval (ontaarde boom)
* geneste tabellen: $O(1+\log(\alpha))$

oef.4 * $d_1 d_2 d_3$ zorgt voor funneling: stel dat je in een bepaald jaar begint te nummeren bij 95000, dan zullen alle 3000 studenten op posities 950 tot 980 terechtkomen!

* dit is de "middle square method", die de eigenschap heeft dat ze opeenvolgende keys mooi zal spreiden. dat is hier misschien overkill.

* $d_3 d_4 d_5$ is hier de beste keuze: aangezien rolnummers sequentieel worden toegekend zullen we een vrijwel uniforme verdeling krijgen.

oef.5 * $d_5 d_6$ aangezien lokalen sequentieel genummerd worden vanaf 1, en een verdieping wellicht meestal geen 100 lokalen telt, krijgen we wellicht funneling links in de tabel

* dit is de folding method
* dit is de multiplication method

cf. cursustekst