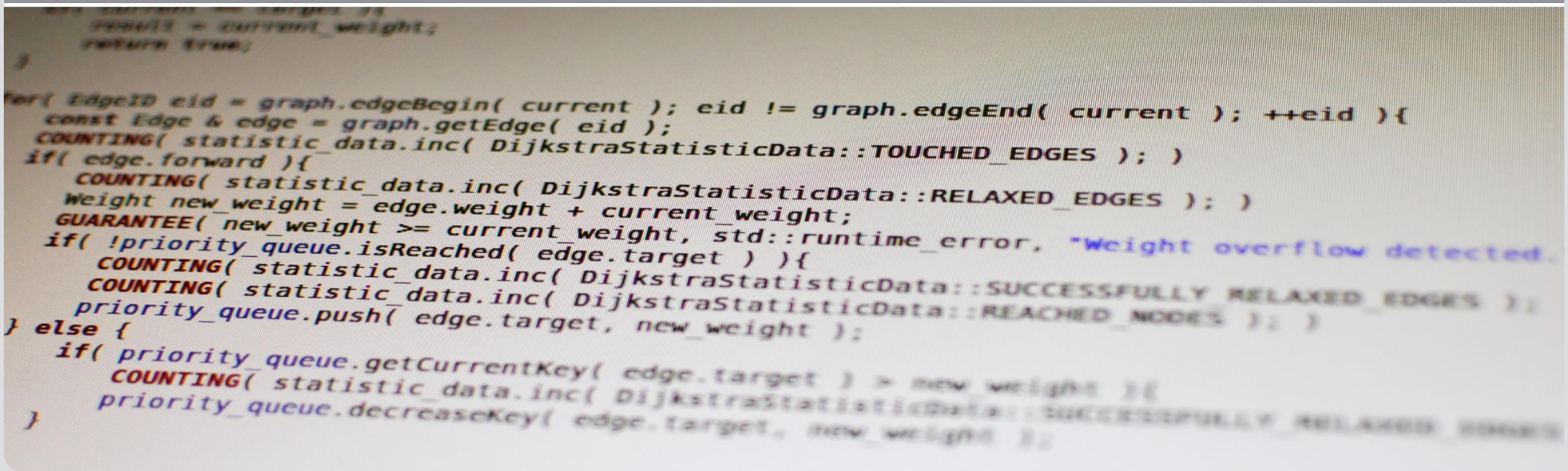
Karlsruher Institut für Technologie

Übung 9 – Algorithmen II

Yaroslav Akhremtsev, Demian Hespe – *yaroslav.akhremtsev@kit.edu, hespe@kit.edu*
Mit Folien von Michael Axtmann

`http://algo2.iti.kit.edu/AlgorithmenII_WS17.php`

Institut für Theoretische Informatik - Algorithmik II



Spezielle *Priority Queues*

monoton, ganzzahlig

Bedingungen

- positive, ganzzahlige Schlüssel
- neue/geänderte Schlüssel $\geq$ minimaler Schlüssel *min*
- maximales Schlüsselinkrement *C*
  (bezogen auf minimalen Schlüssel *min*)

Spezielle *Priority Queues*

monoton, ganzzahlig

Warum das Ganze?

- spezialisierte Datenstruktur $\longrightarrow$ schneller
- Dijkstras Algorithmus

Idee

- speichere Schlüssel in *Buckets* statt Bäumen

Spezielle *Priority Queues*

Dijkstras Algorithmus

Laufzeit Dijkstras Algorithmus

- $T_{Dijkstra} = O(m \cdot T_{decreaseKey} + n \cdot (T_{deleteMin} + T_{insert}))$

amortisierte Laufzeiten

$O(\cdot)$	*Bin. Heap*	*Fib. Heap*	*Bkt. Queue*	*Radix Heap*
$T_{decreaseKey}$	$\log n$	1	1	1
T_{insert}	$\log n$	1	1	$\log C$
$T_{deleteMin}$	$\log n$	$\log n$	C	$\log C$
$T_{Djkstra}$	$(m+n)\log n$	$m + n\log n$	$m + nC$	$m + n\log C$

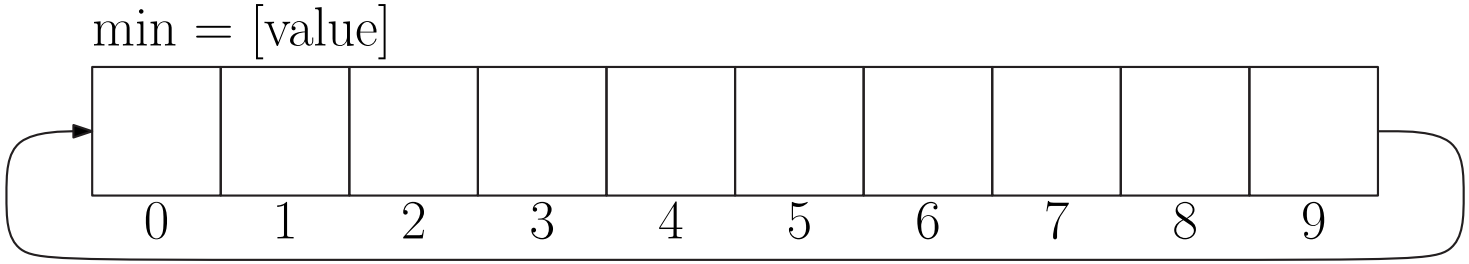
Bucket Queues

Aufbau:

- zirkuläre Liste aus Buckets $B[\cdot]$
- Variable mit minimalem Schlüssel *min*
  (der in die Datenstruktur aufgenommen werden kann)
- *min*: letzter `deleteMin`-Schlüssel, initialisiert mit 0

gegeben:

- max. Schlüsselinkrement $C \longrightarrow$ # Buckets $|B| = C + 1$



$$C = 9 \longrightarrow |B| = C + 1 = 10$$

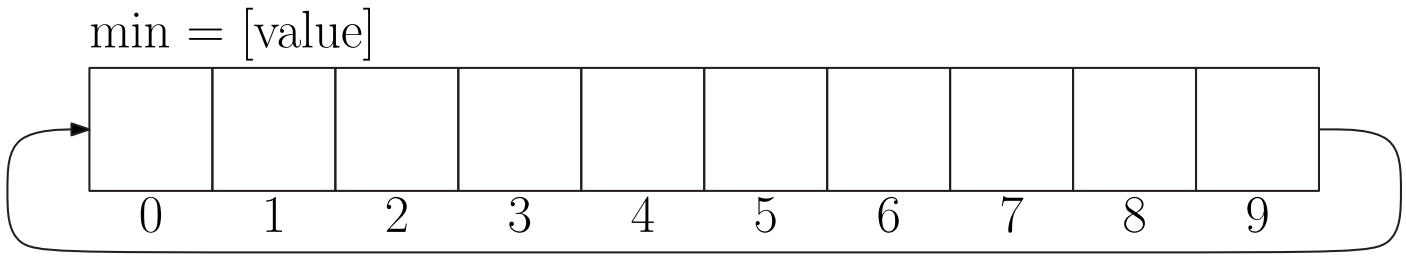
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



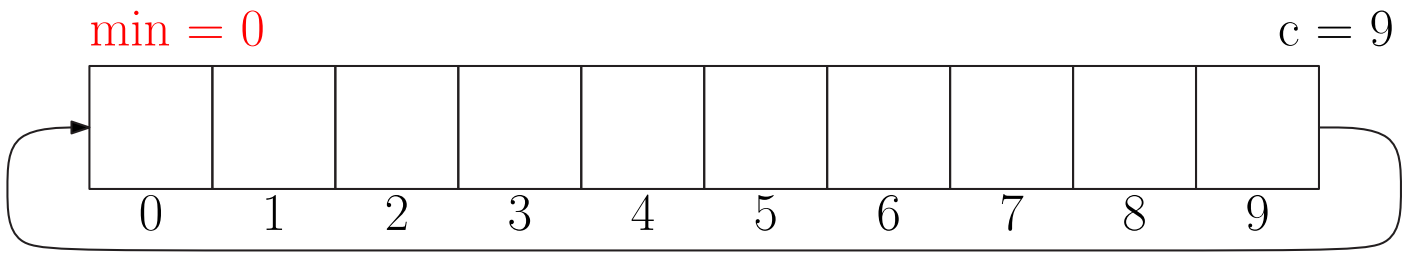
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



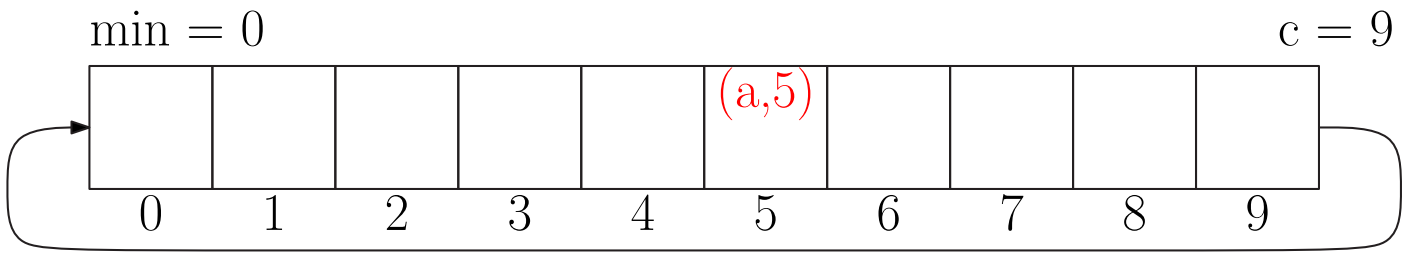
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



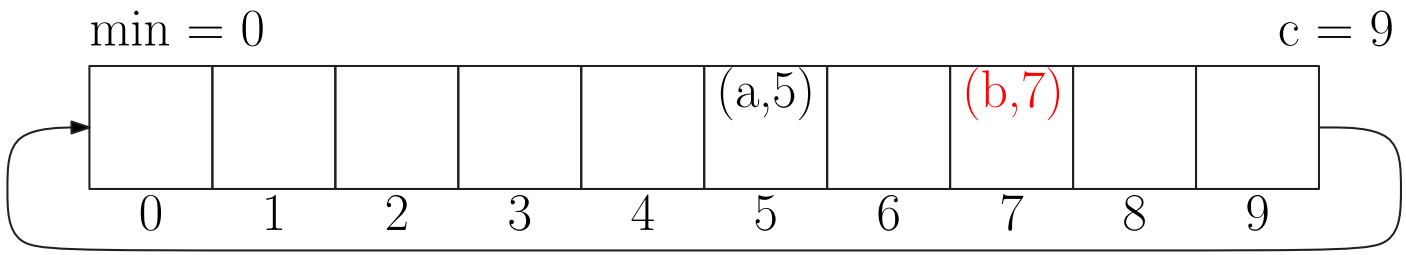
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



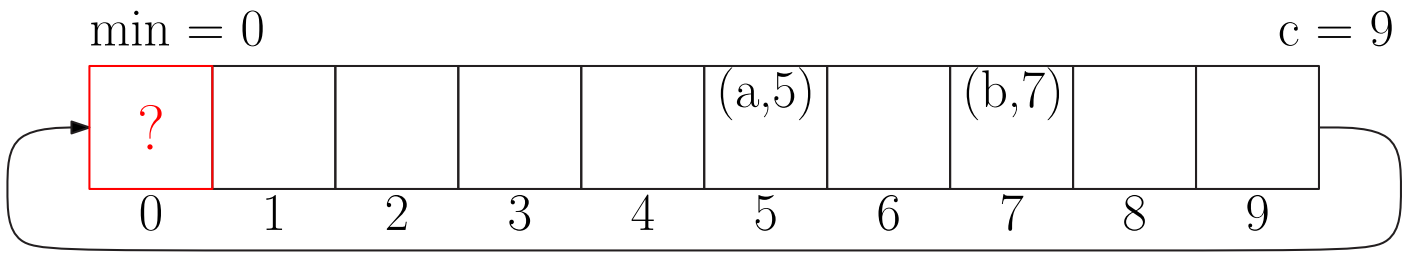
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



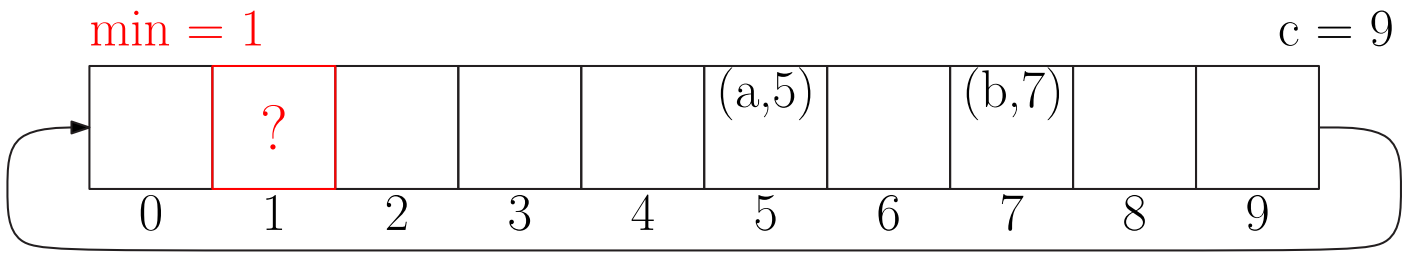
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



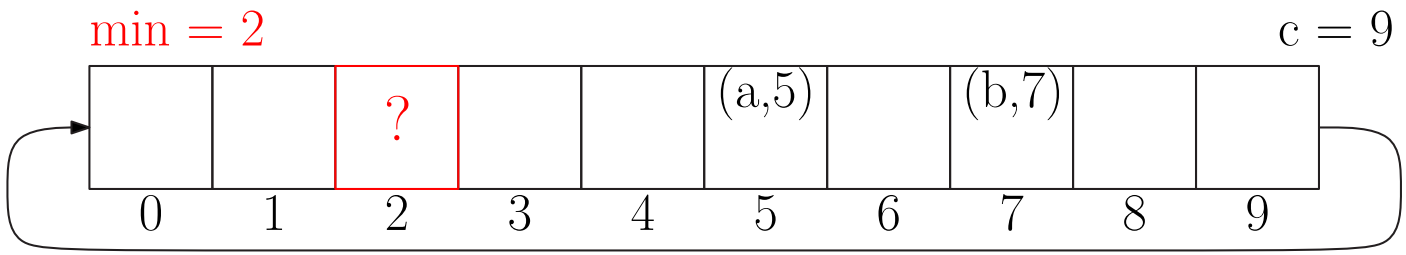
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



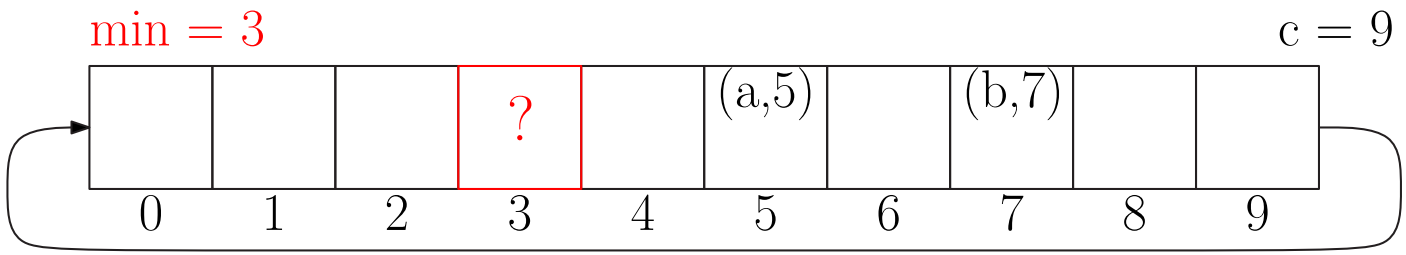
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



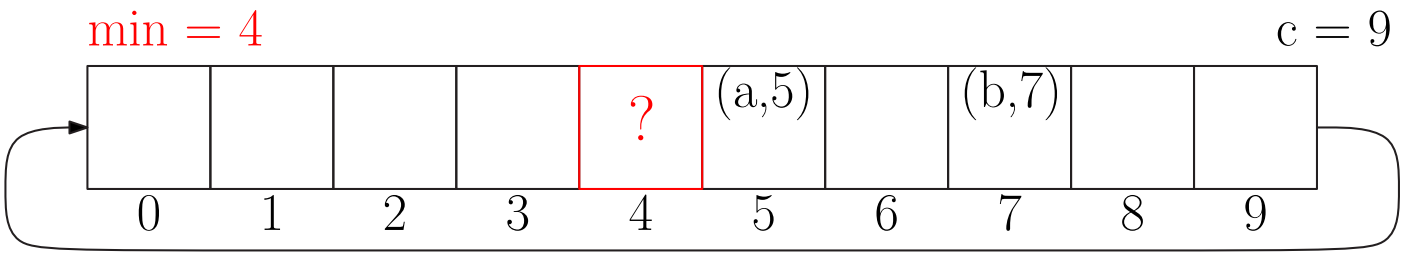
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



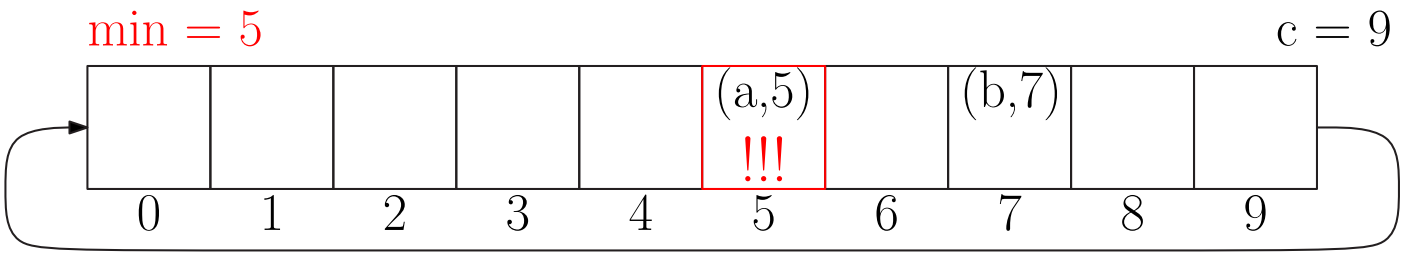
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



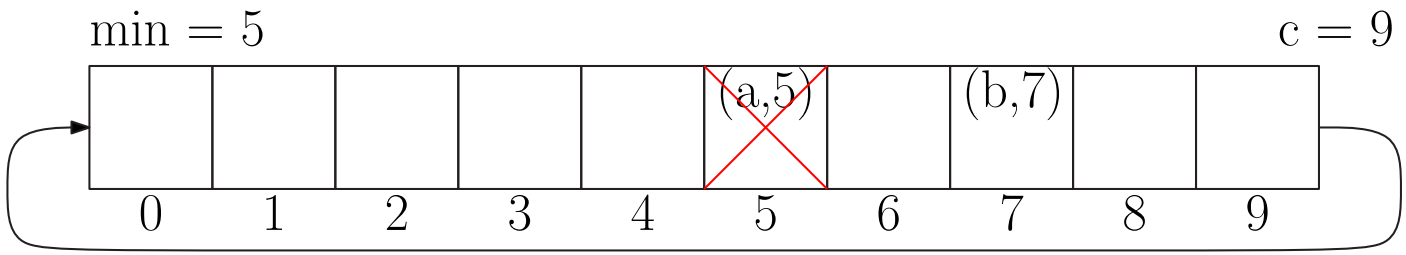
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



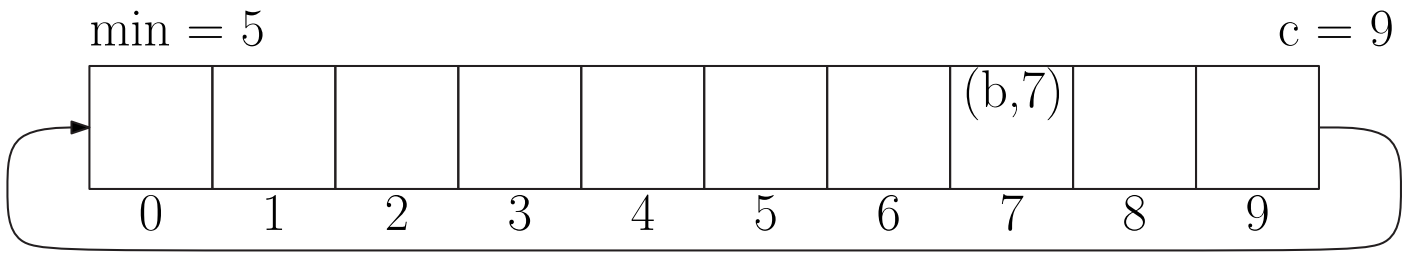
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



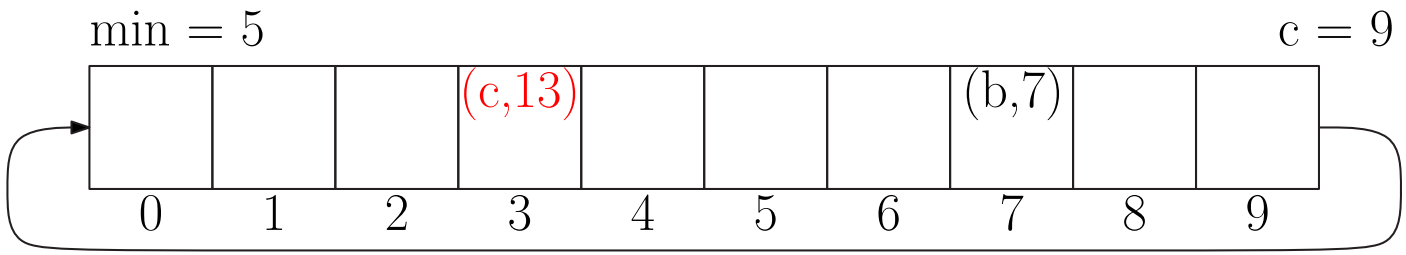
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



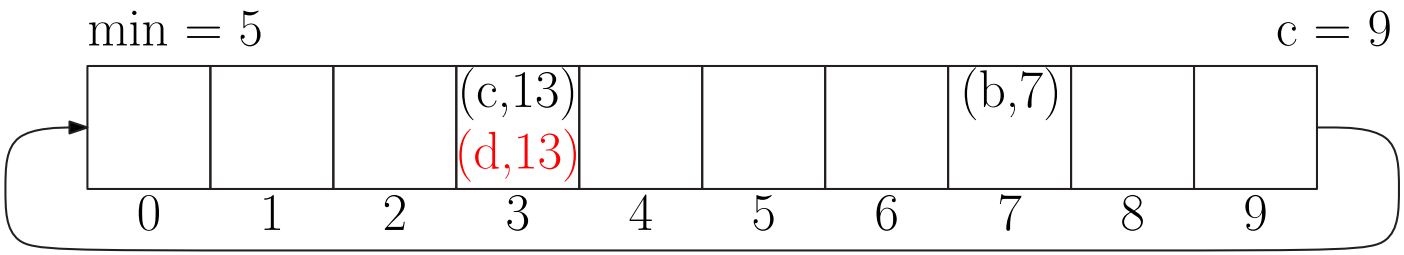
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



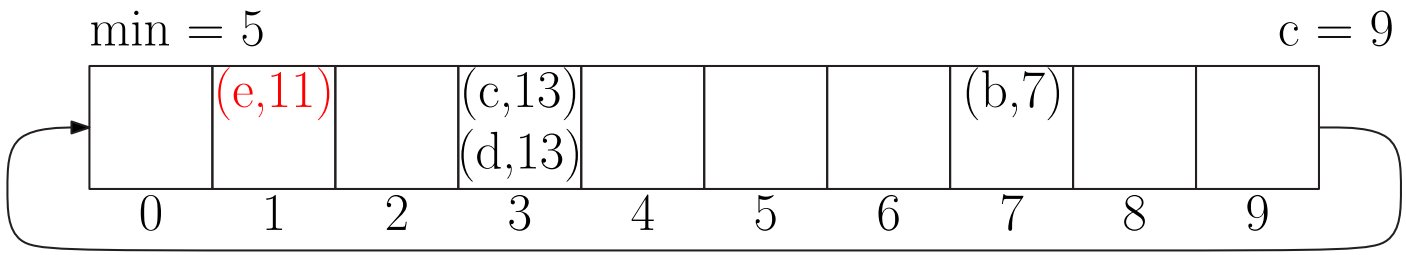
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



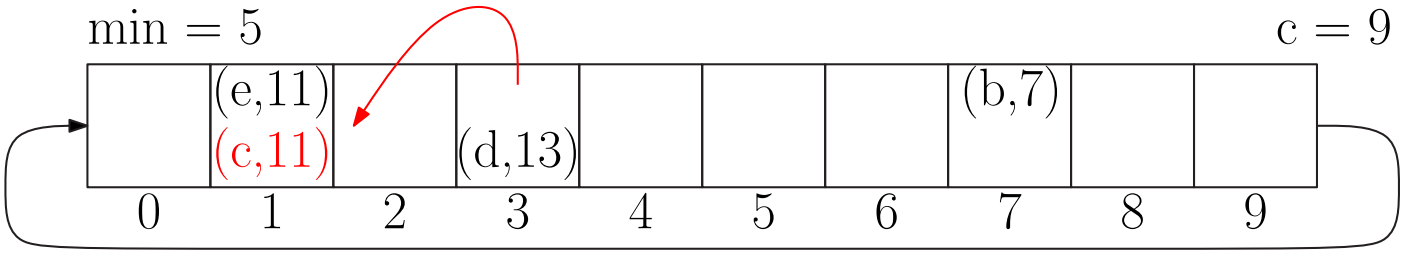
Bucket Queues

Ablauf:

`insert`:	Einfügen in Bucket $B[key \bmod (C+1)]$	$O(1)$
`deleteMin`:	Entfernen aus Bucket $B[min \bmod (C+1)]$	$O(C)$
	(bzw. aus erstem nicht-leeren Folgebucket)	
`decreaseKey`:	Verschieben von altem in neuen Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



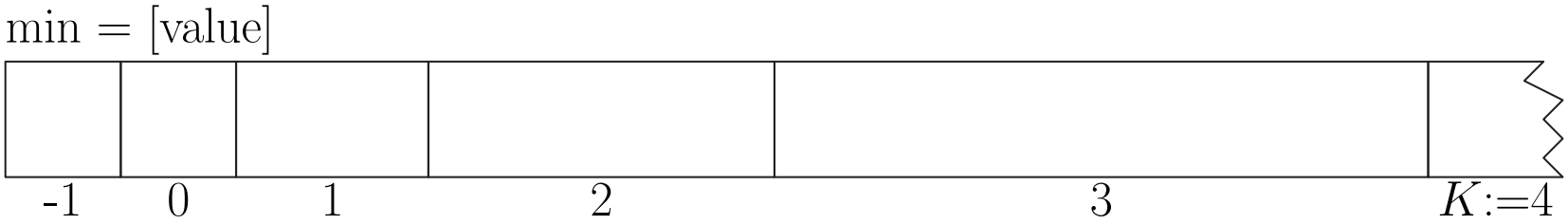
Radix Heaps

Aufbau:

- Liste aus logarithmischer Anzahl Buckets $B[\cdot]$
- Variable mit minimalem Schlüssel *min*
  (der in die Datenstruktur aufgenommen werden kann)

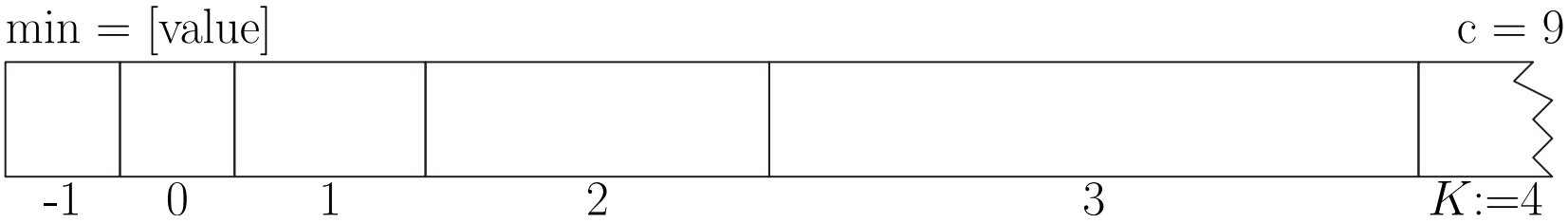
gegeben:

- max. Schlüsselinkrement $C \longrightarrow$ # Buckets $|B| = K + 2$
  $= (\lfloor \log C \rfloor + 1) + 2$



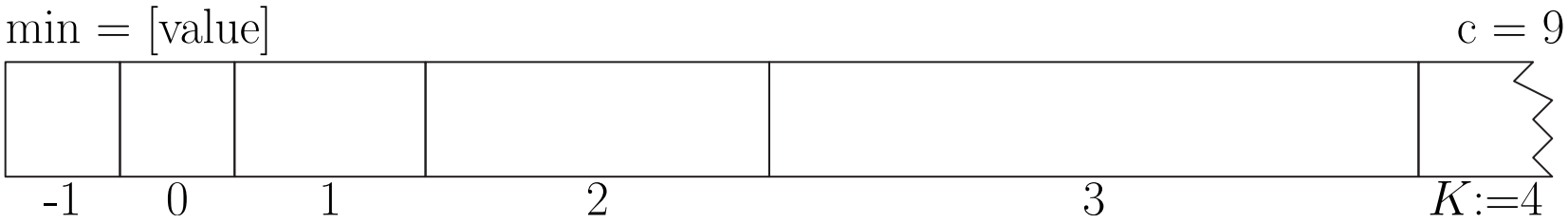
$$C = 9 \longrightarrow K + 2 = (\lfloor \log C \rfloor + 1) + 2 = (3 + 1) + 2 = 6$$

Radix Heaps



Welche Schlüssel kommen in welches Bucket?

Radix Heaps



Welche Schlüssel kommen in welches Bucket?

Bucket i: hält Elemente mit $i = \min(\text{msd}(min, key), K)$

$\rightarrow$ Bucket K ist *overflow* Bucket

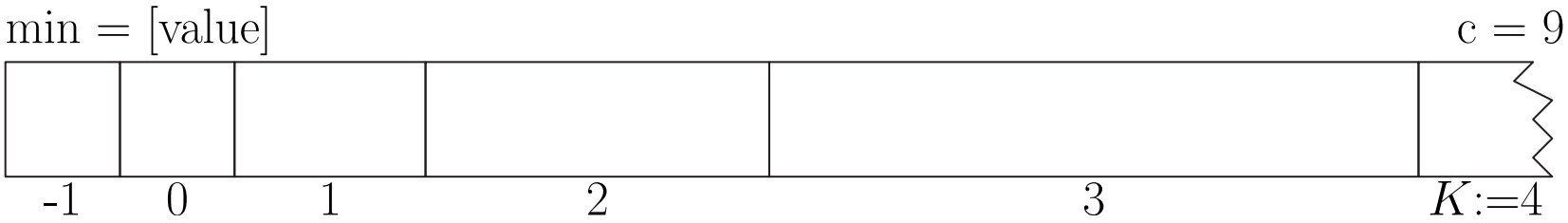
$\rightarrow$ höchstwertiges unterschiedliche Bit ist i

(bzw. -1 wenn $key = min$)

$\rightarrow$ $\max(1,\ 2^i)$ verschiedene Schlüssel

$\rightarrow$ leer, wenn Bit i von *min* gesetzt

Radix Heaps



Welche Schlüssel kommen in welches Bucket?

Bucket i: hält Elemente mit $i = \min(\text{msd}(min, key), K)$

$\rightarrow$ Bucket K ist *overflow* Bucket

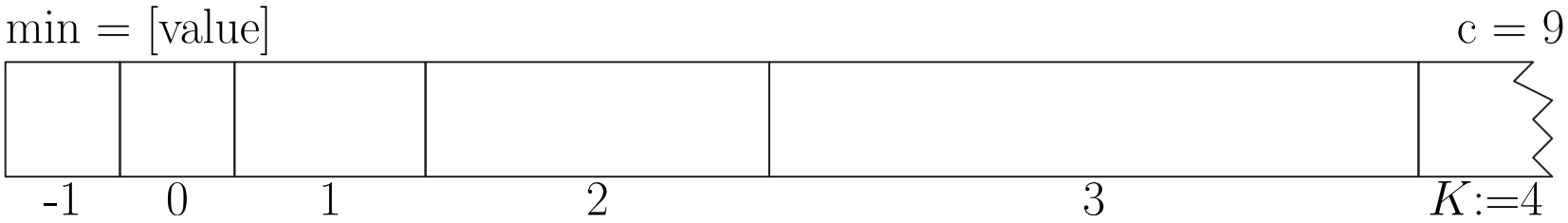
$\rightarrow$ höchstwertiges unterschiedliche Bit ist i

(bzw. -1 wenn $key = min$)

$\rightarrow \max(1, 2^i)$ verschiedene Schlüssel

$\rightarrow$ leer, wenn Bit i von min gesetzt

Radix Heaps



Welche Schlüssel kommen in welches Bucket?

Bucket i: hält Elemente mit $i = \min(\text{msd}(min, key), K)$

$\rightarrow$ Bucket K ist *overflow* Bucket

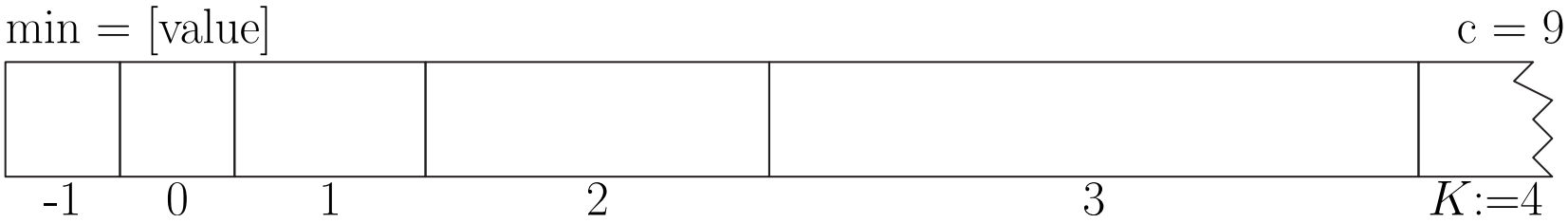
$\rightarrow$ höchstwertiges unterschiedliche Bit ist i

(bzw. -1 wenn $key = min$)

$\rightarrow \max(1, 2^i)$ verschiedene Schlüssel

$\rightarrow$ leer, wenn Bit i von min gesetzt

Radix Heaps



Welche Schlüssel kommen in welches Bucket?

Bucket i: hält Elemente mit $i = \min(\text{msd}(\textit{min}, \textit{key}), K)$

$\rightarrow$ Bucket K ist *overflow* Bucket

$\rightarrow$ höchstwertiges unterschiedliche Bit ist i

(bzw. -1 wenn $\textit{key} = \textit{min}$)

$\rightarrow \max(1,\ 2^i)$ verschiedene Schlüssel

$\rightarrow$ leer, wenn Bit i von *min* gesetzt

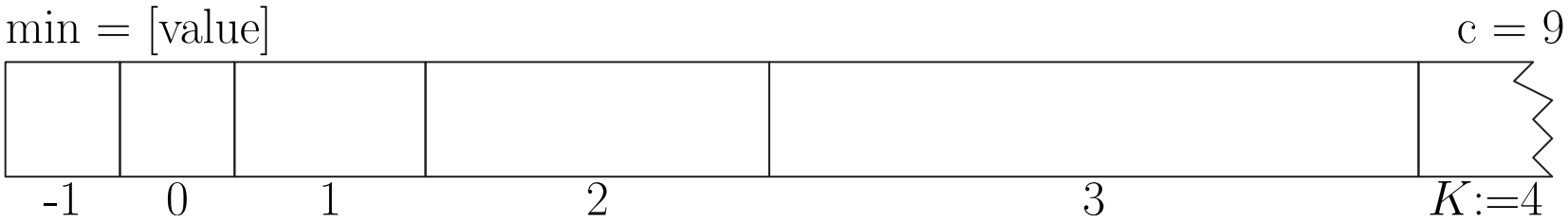
Beispiele

$\textit{min} := \texttt{00001000}\ (08)$

$\text{msd}(\texttt{00001010},\ \texttt{01001000}) = 6$

$\text{msd}(\texttt{00001000},\ \texttt{00001010}) = 1$

Radix Heaps



Welche Schlüssel kommen in welches Bucket?

Bucket i: hält Elemente mit $i = \min(\text{msd}(min, key), K)$

$\rightarrow$ Bucket K ist *overflow* Bucket

$\rightarrow$ höchstwertiges unterschiedliche Bit ist i

(bzw. -1 wenn $key = min$)

$\rightarrow \max(1, 2^i)$ verschiedene Schlüssel

$\rightarrow$ leer, wenn Bit i von *min* gesetzt

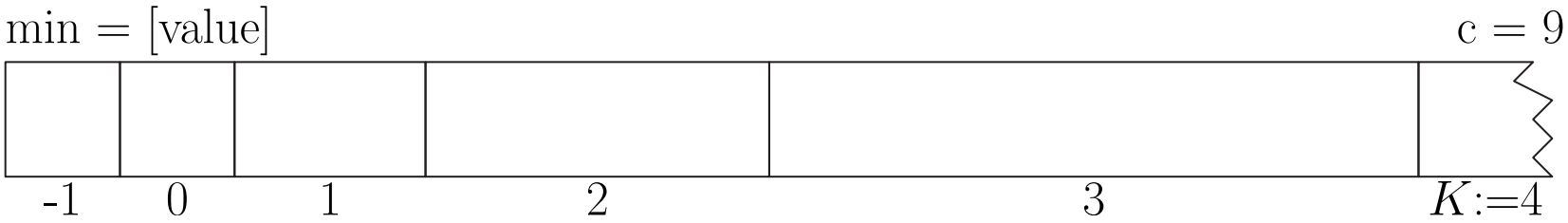
Beispiele

min := 00001000 (08)

$\text{msd}(00001000, 01******) = 6 \longrightarrow 2^6 = 64$ Werte

$\text{msd}(00001000, 0000101*) = 1 \longrightarrow 2^1 = 2$ Werte

Radix Heaps



Welche Schlüssel kommen in welches Bucket?

Bucket i: hält Elemente mit $i = \min(\text{msd}(min, key), K)$

$\rightarrow$ Bucket K ist *overflow* Bucket

$\rightarrow$ höchstwertiges unterschiedliche Bit ist i

(bzw. -1 wenn $key = min$)

$\rightarrow \max(1,\ 2^i)$ verschiedene Schlüssel

$\rightarrow$ leer, wenn Bit i von *min* gesetzt

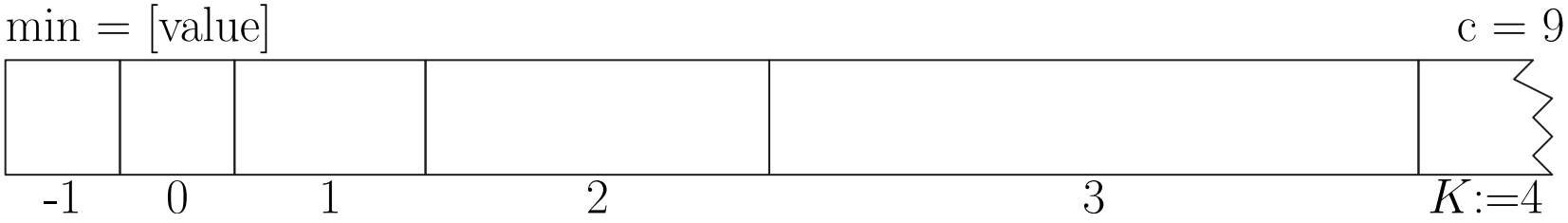
Beispiele

min := `00001000` (08)

$\text{msd}(\texttt{00001000},\ \texttt{00000110}) = 3 \longrightarrow$ kann nicht passieren, da $6 < 8$

$\text{msd}(\texttt{00001000},\ \texttt{00000100}) = 3 \longrightarrow$ kann nicht passieren, da $4 < 8$

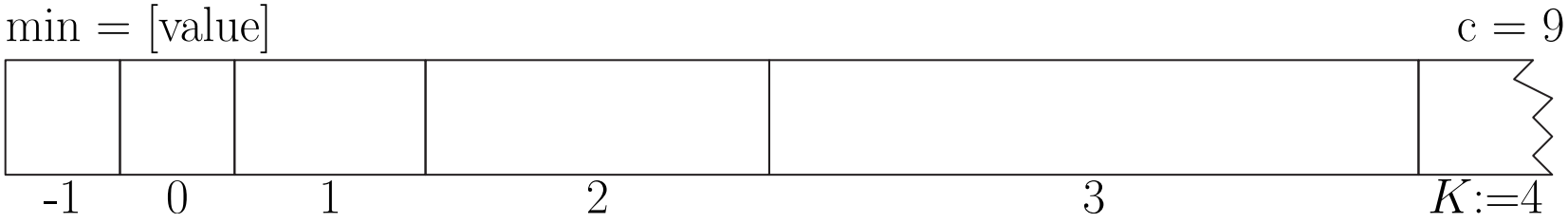
Radix Heaps



Welche Schlüssel kommen in welches Bucket?

	Bucket $B[\cdot]$					
min	-1	0	1	2	3	4
1000 (08)	8	9	10..11	12..15	-	16..8+C
1010 (10)	10	11	-	12..15	-	16..10+C
1111 (15)	15	-	-	-	-	16..15+C
1000100 (68)	68	69	70..71	-	72..79 ?	80..69+C ?

Radix Heaps



Welche Schlüssel kommen in welches Bucket?

			Bucket $B[\cdot]$			
min	-1	0	1	2	3	4
1000 (08)	8	9	10..11	12..15	-	16..8+C
1010 (10)	10	11	-	12..15	-	16..10+C
1111 (15)	15	-	-	-	-	16..15+C
1000100 (68)	68	69	70..71	-	72..77 !	- !

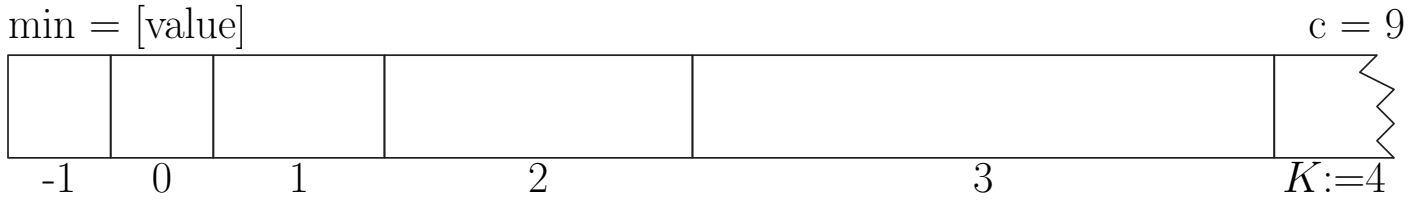
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



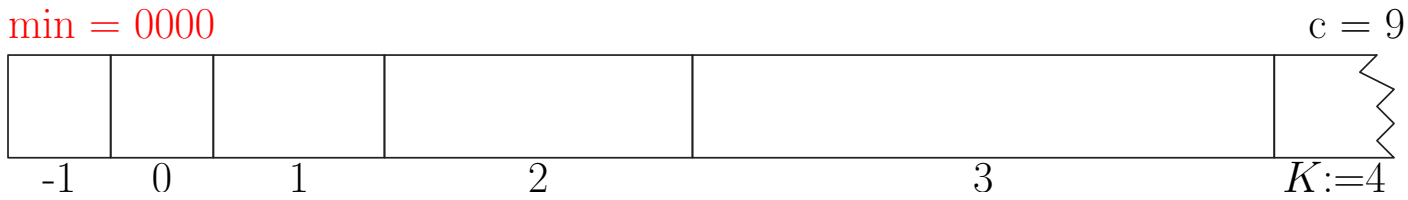
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



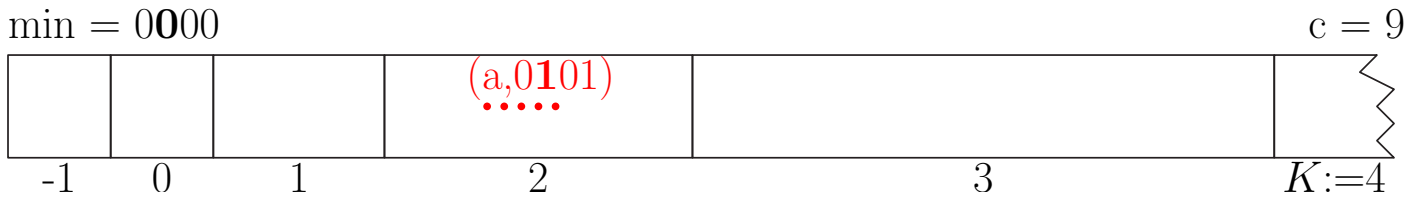
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



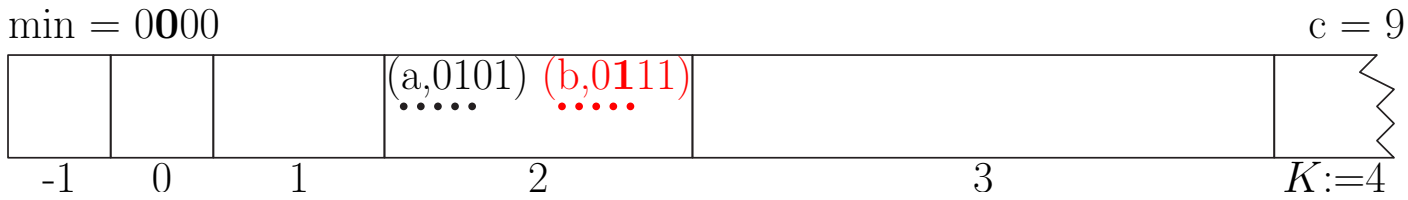
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



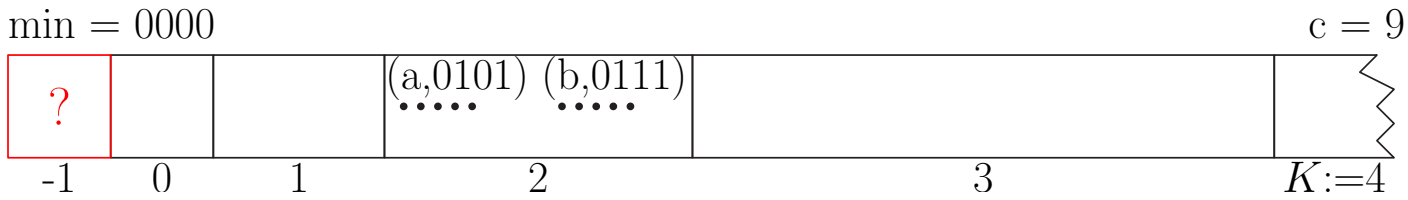
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



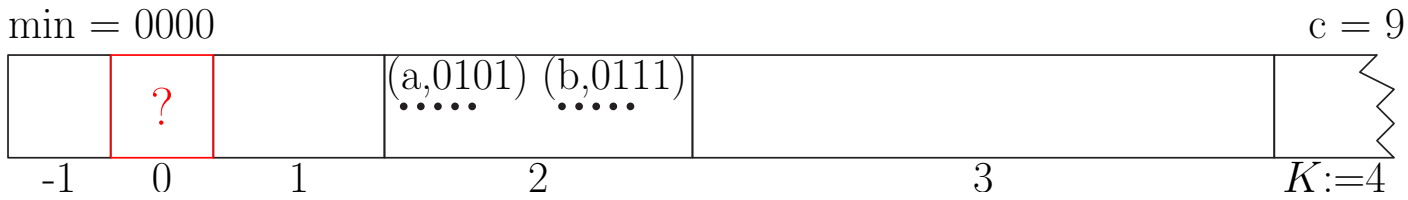
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



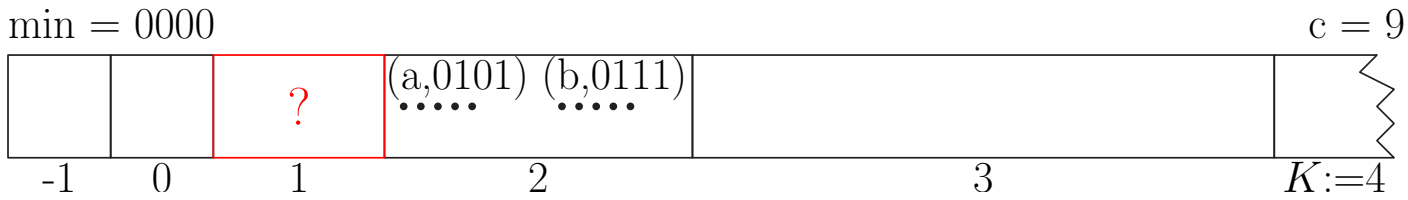
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



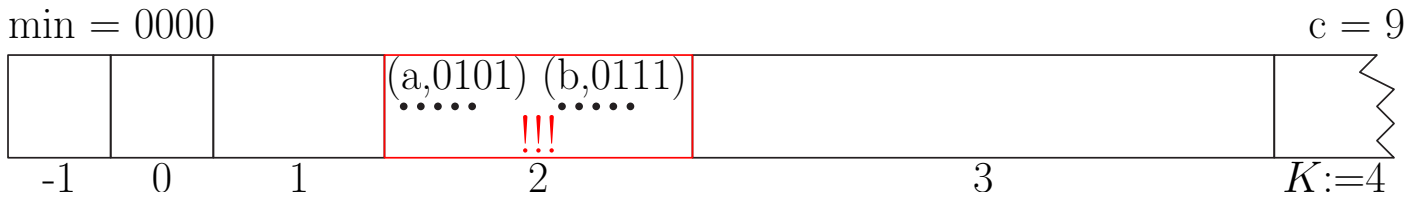
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



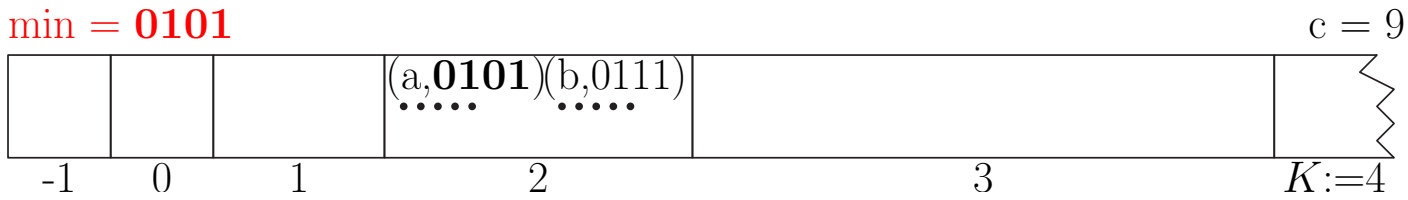
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



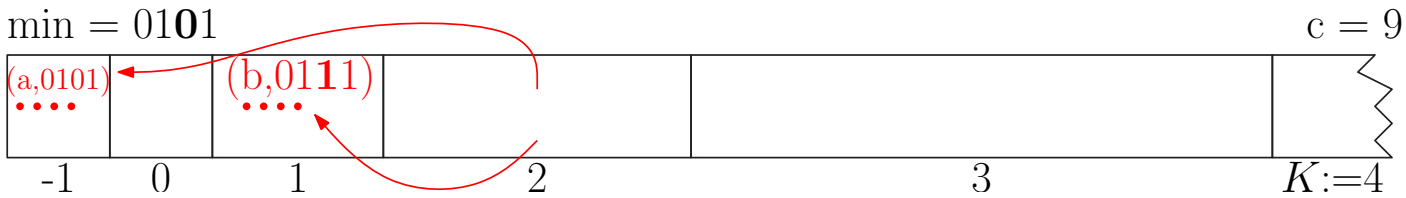
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



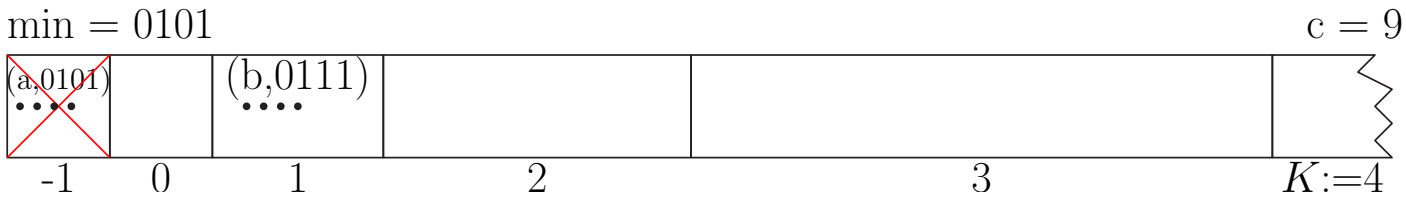
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



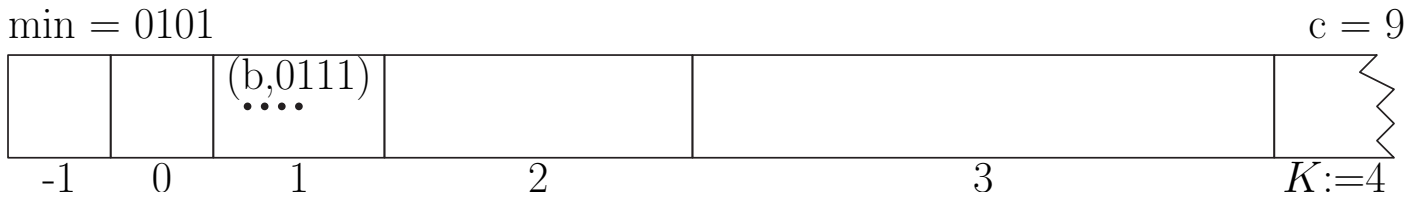
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



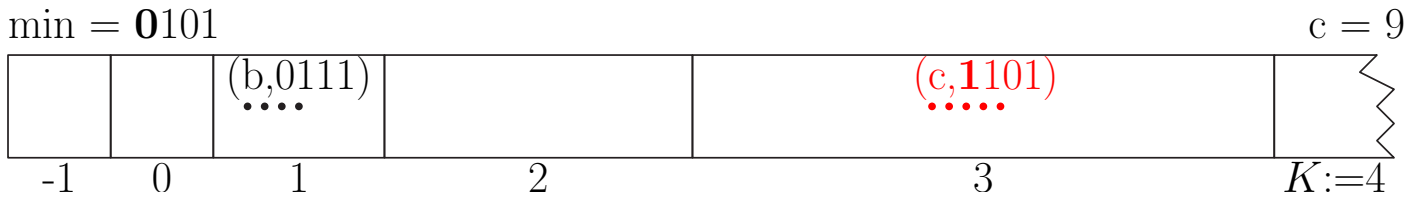
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



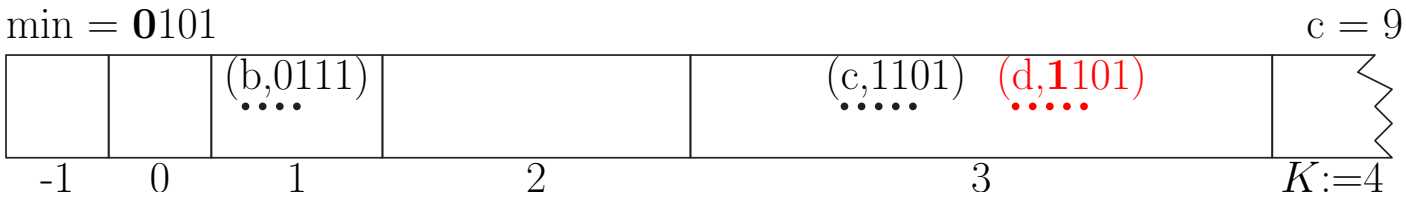
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



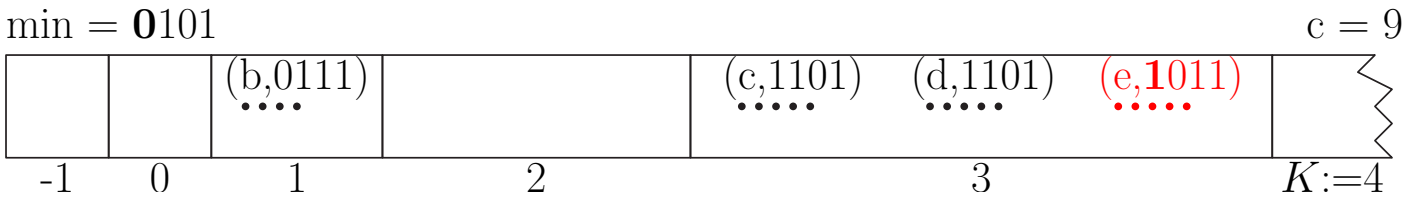
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



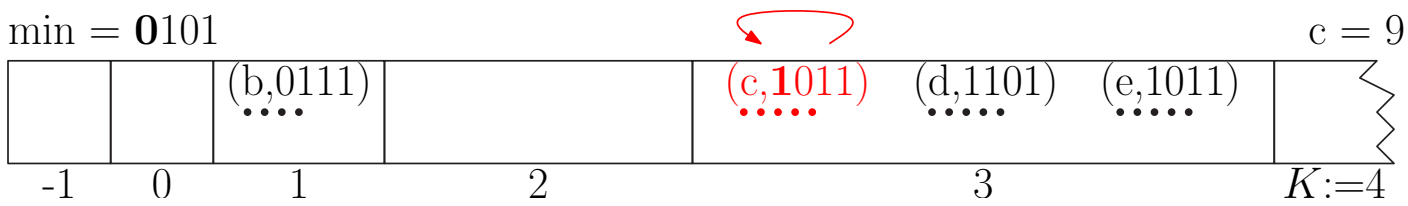
Radix Heaps

Ablauf und Laufzeiten (amortisiert):

`insert`:	Einfügen in Bucket $B[\min(\text{msd}(\mathit{min}, \mathit{key}), K)]$	$O(\log C)$
`deleteMin`:	$\mathit{min} =$ minimaler Schlüssel (aus Bucket i), Elemente aus Bucket i verschieben, Entfernen aus Bucket $B[-1]$	$O(\log C)$
`decreaseKey`:	Verschieben von altem in neues Bucket	$O(1)$

Beispiel:

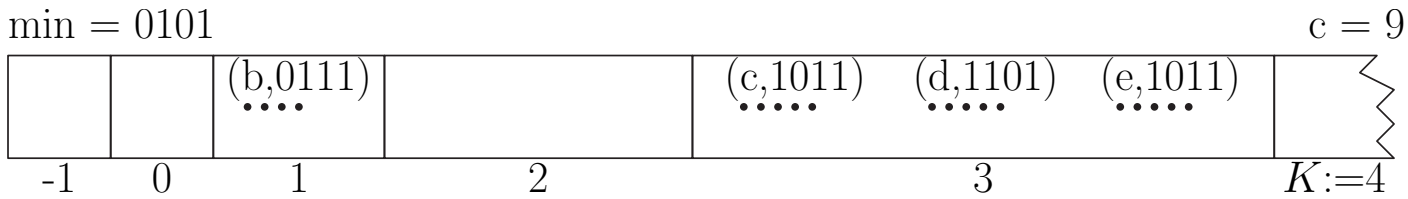
- `insert(a,5)`
- `insert(b,7)`
- `deleteMin()`
- `insert(c,13)`
- `insert(d,13)`
- `insert(e,11)`
- `decreaseKey(c,11)`



Radix Heaps

Beispiel (fortgesetzt):

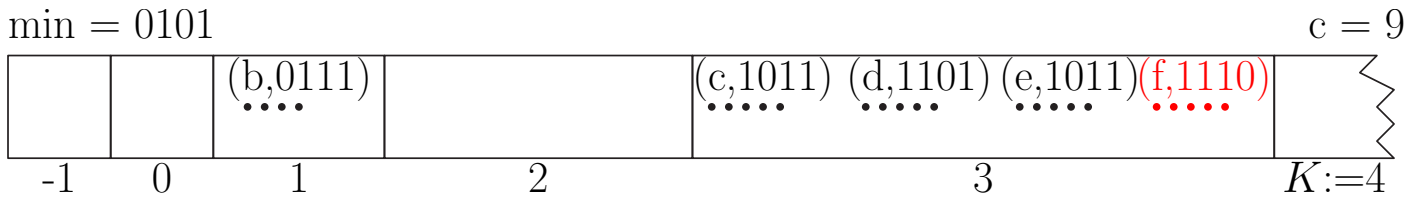
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

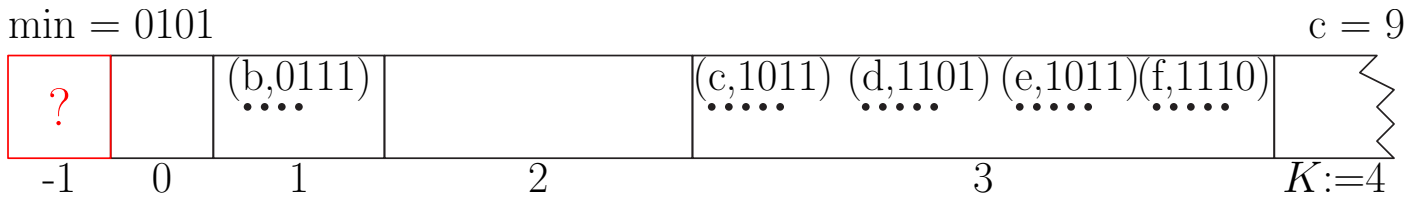
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

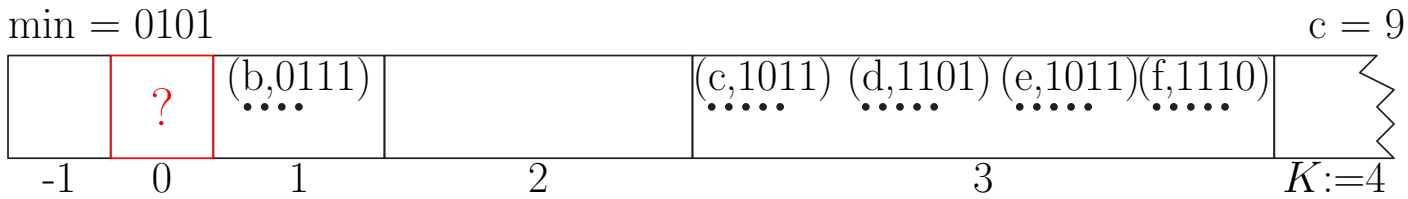
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

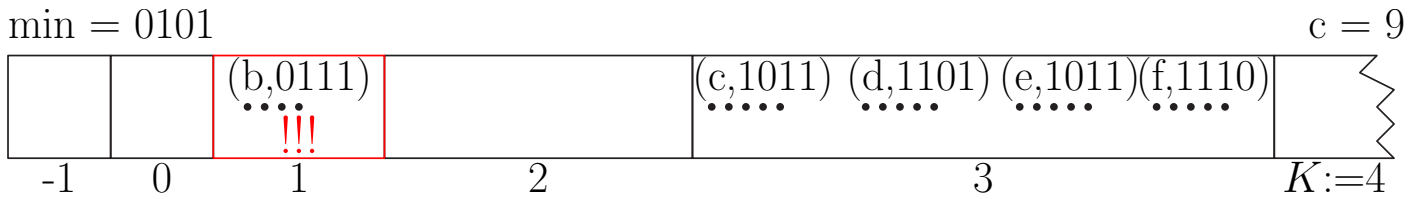
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

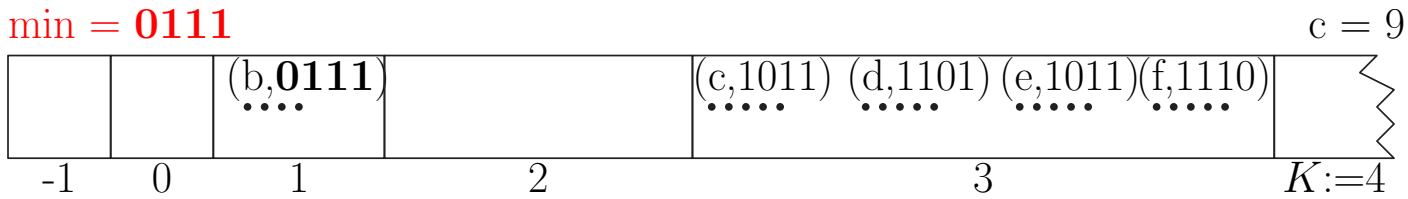
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

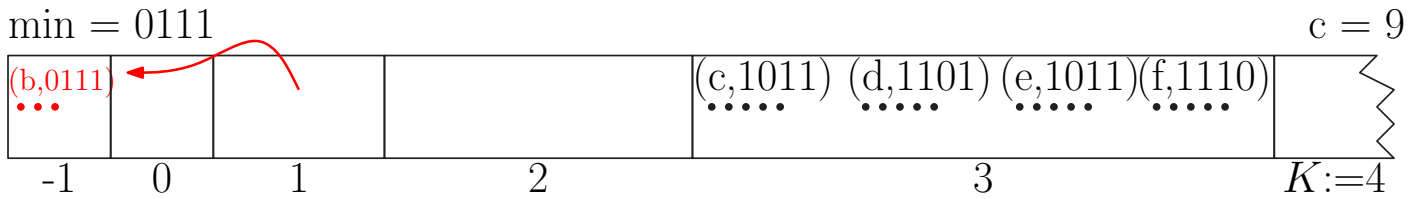
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

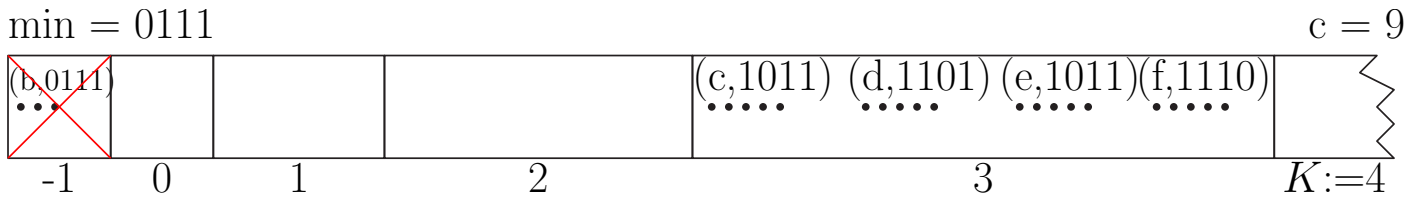
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

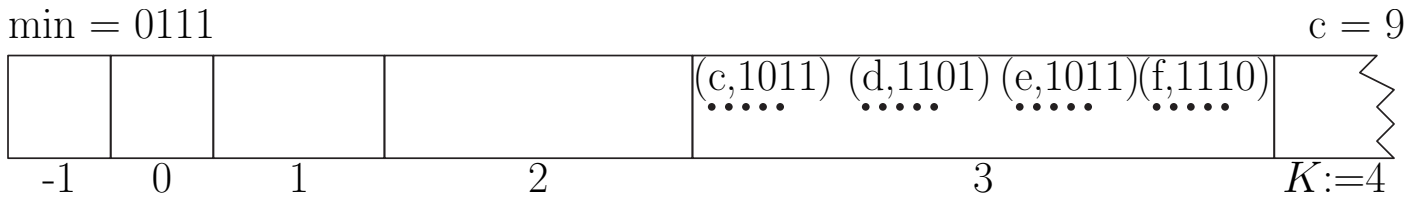
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

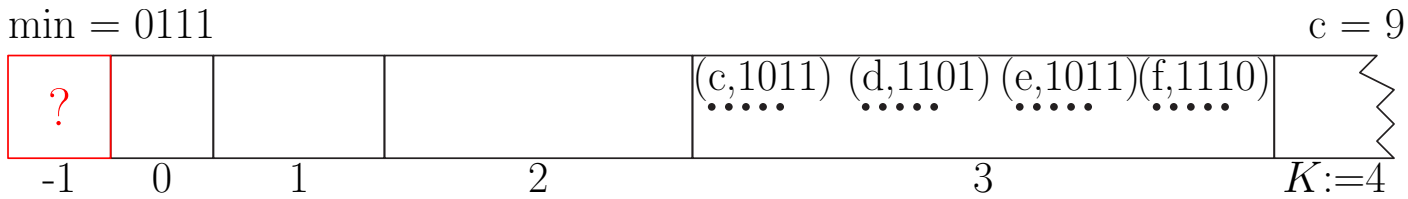
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

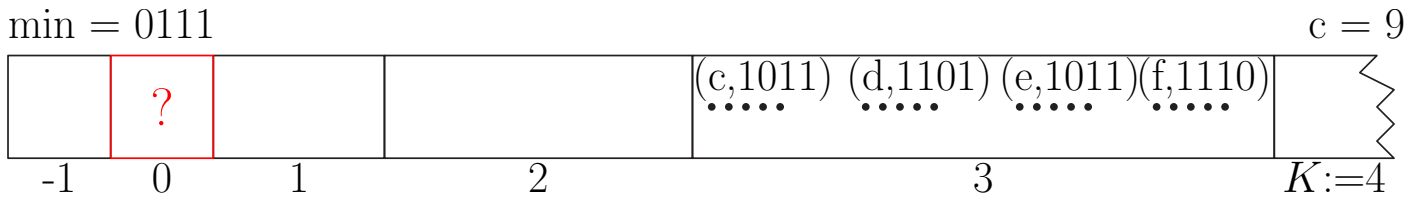
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

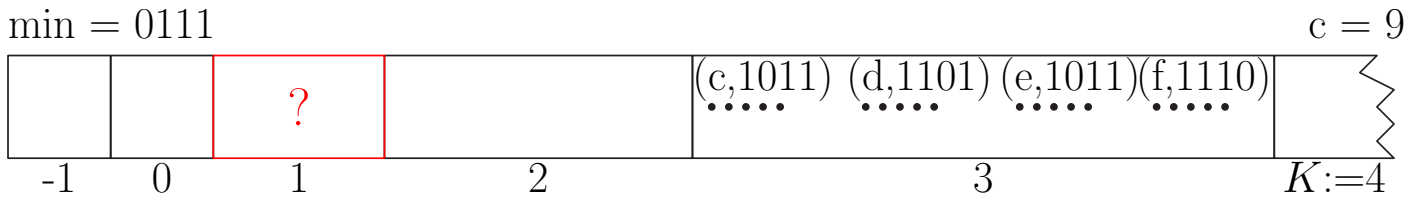
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

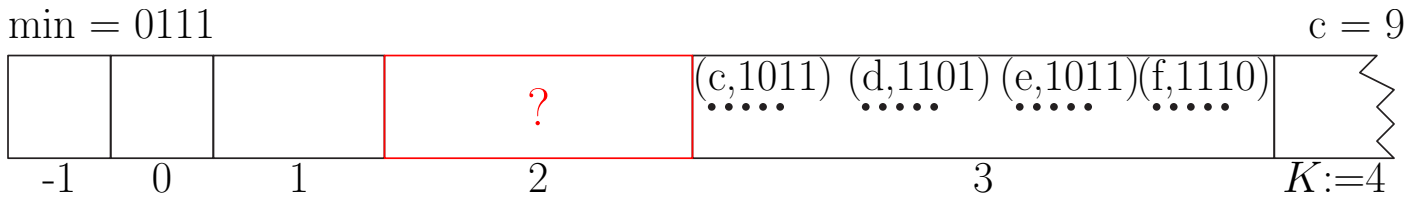
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

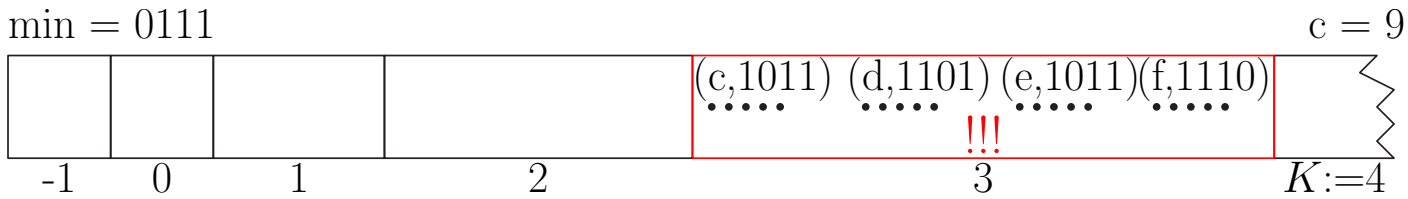
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

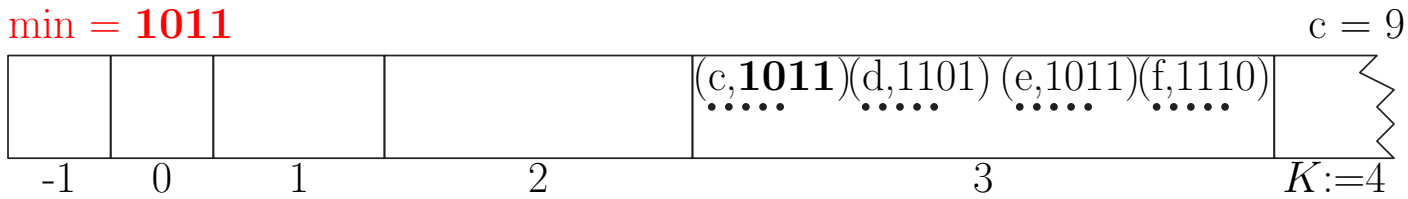
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

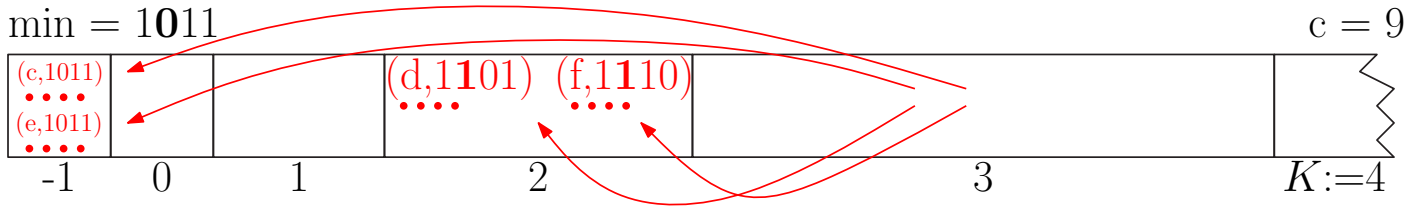
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

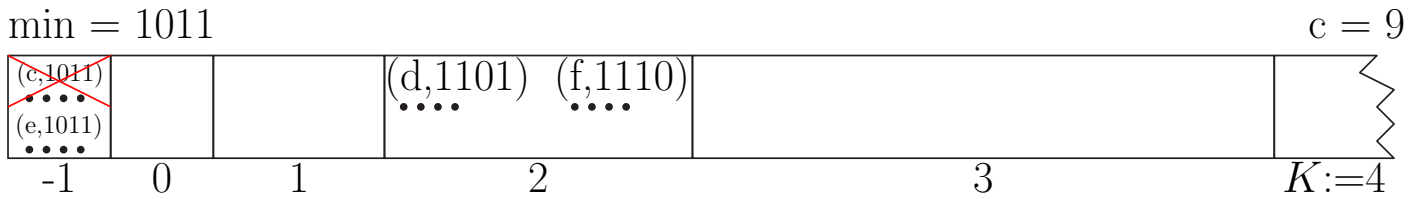
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

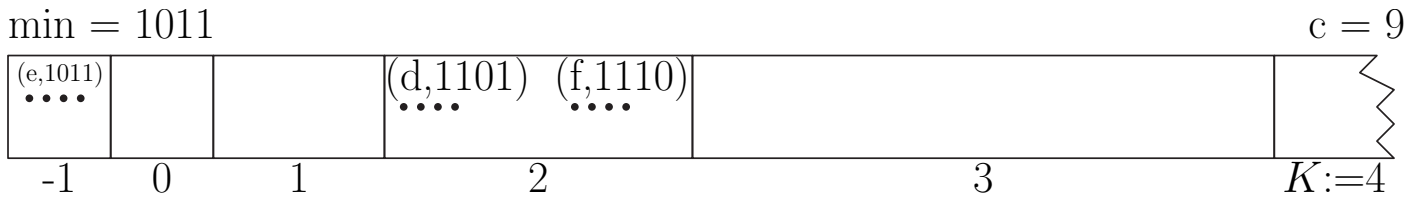
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

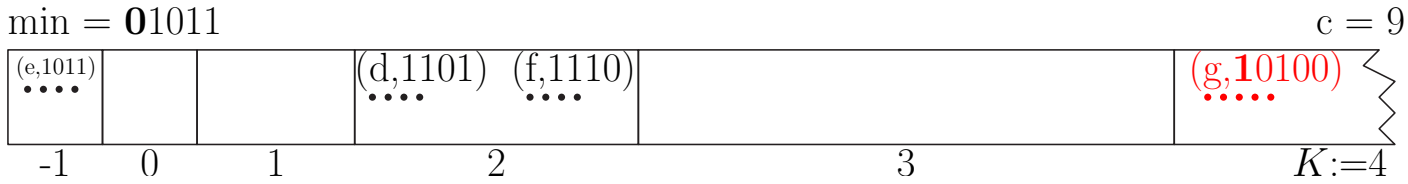
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

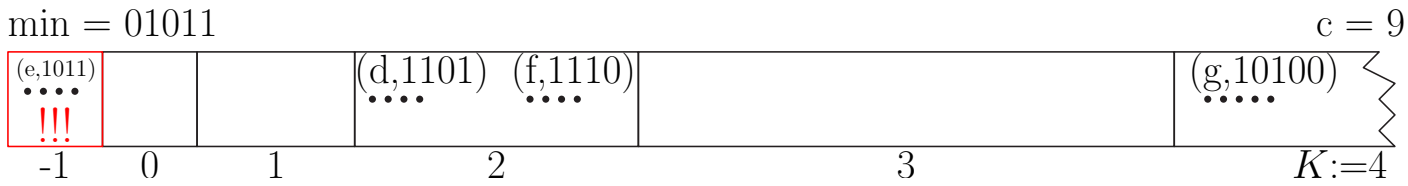
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

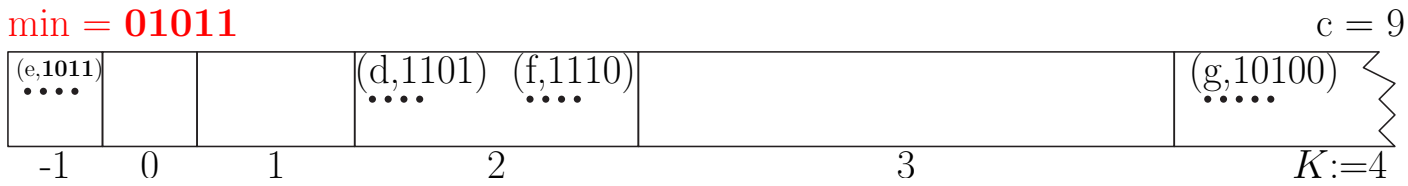
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

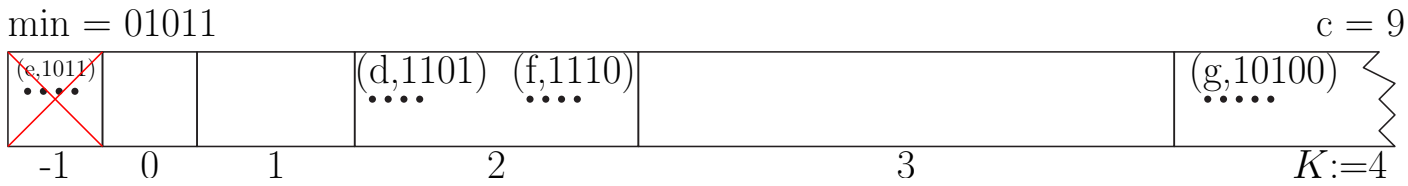
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

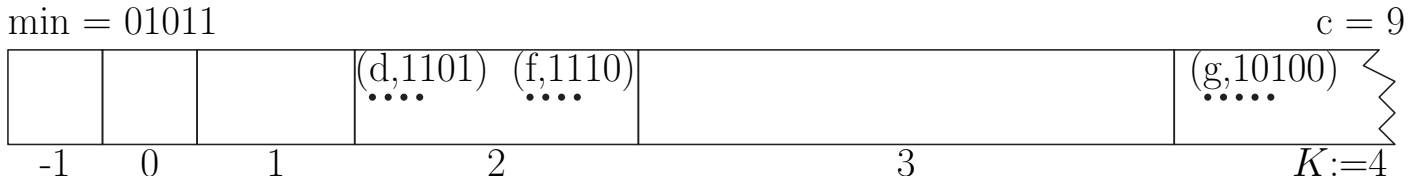
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

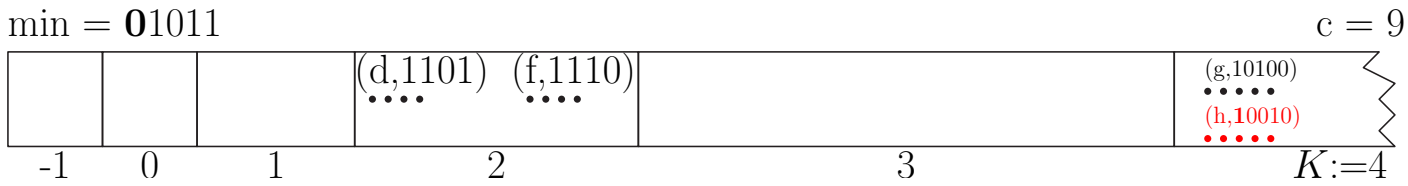
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

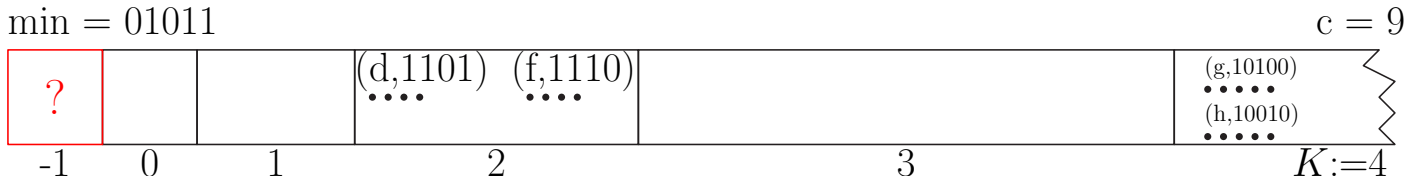
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

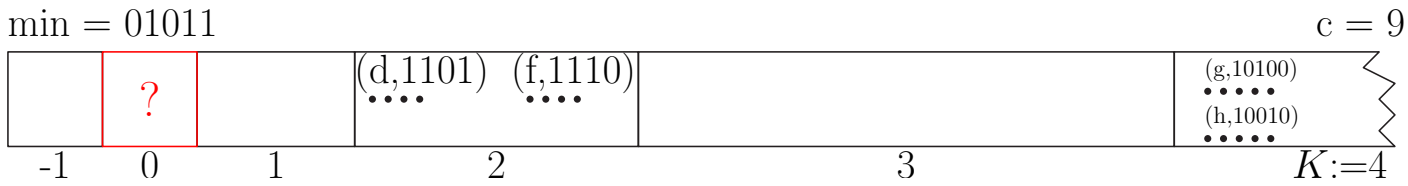
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

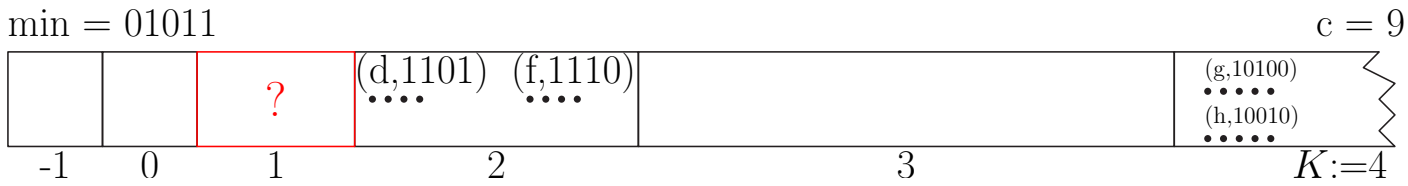
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

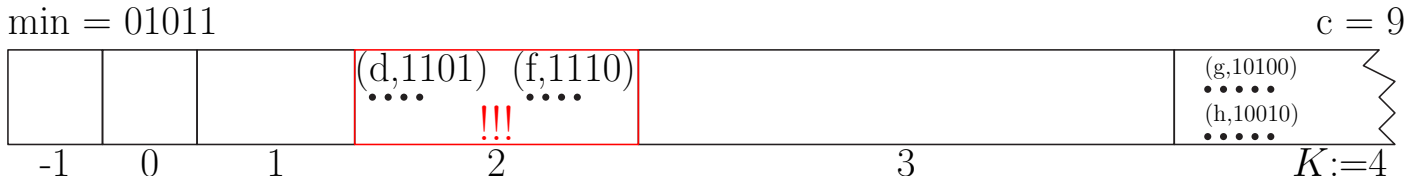
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

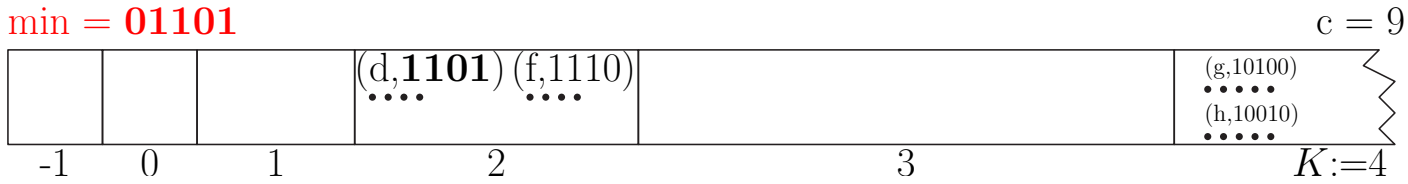
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

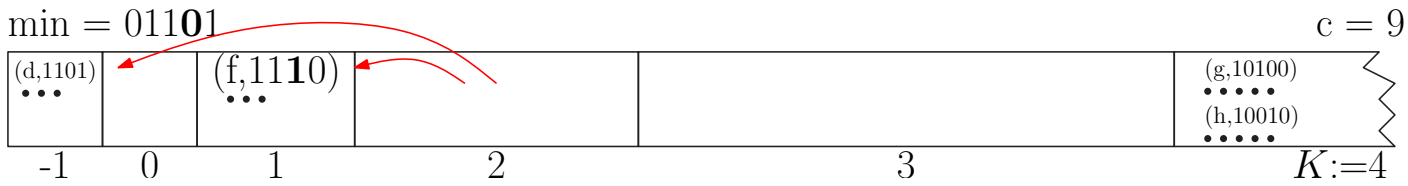
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

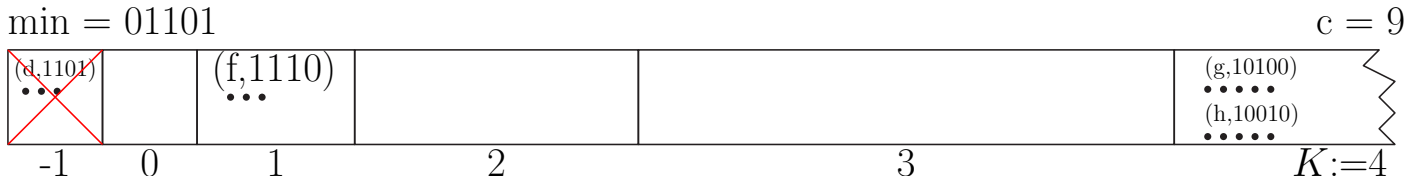
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

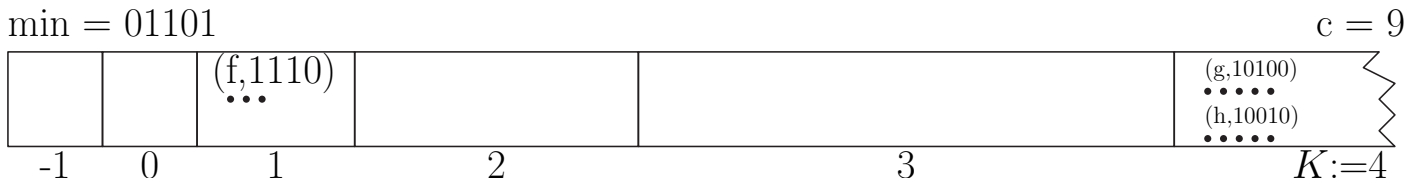
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

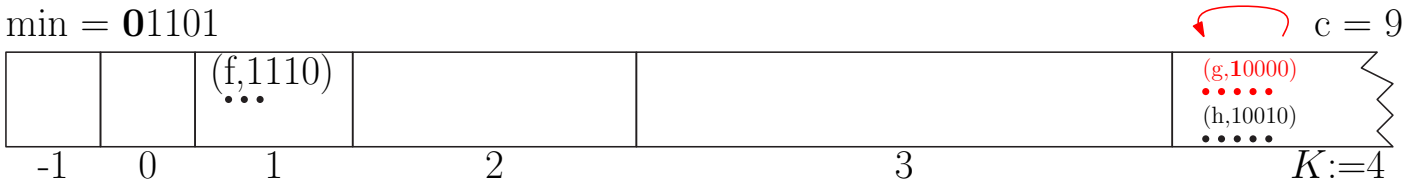
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

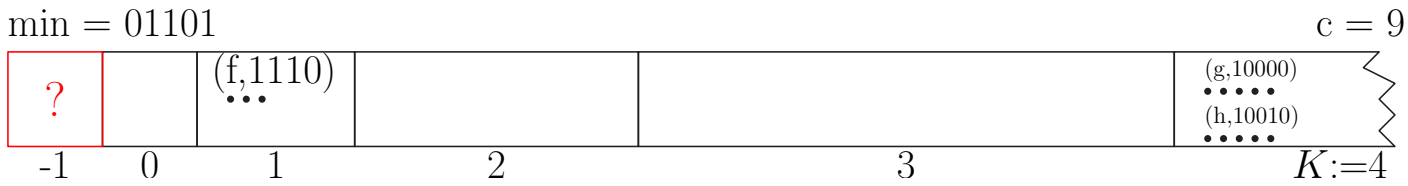
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

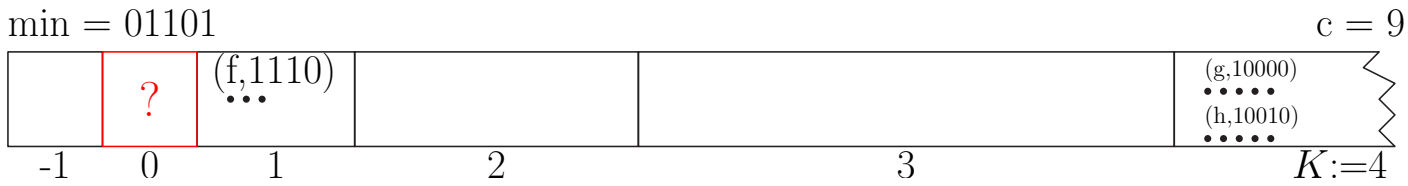
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

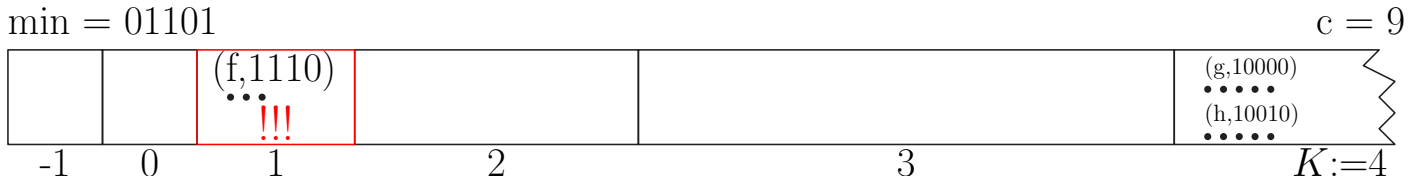
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

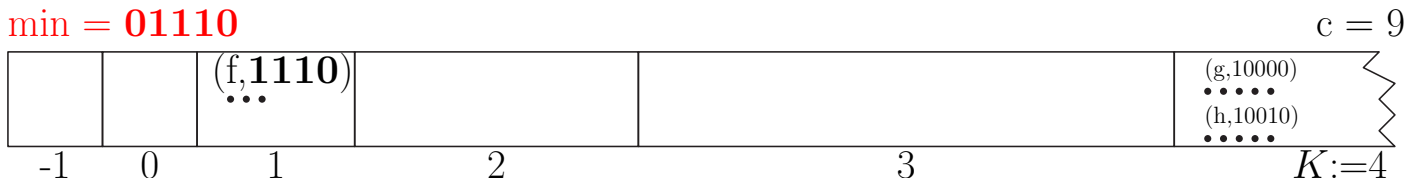
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

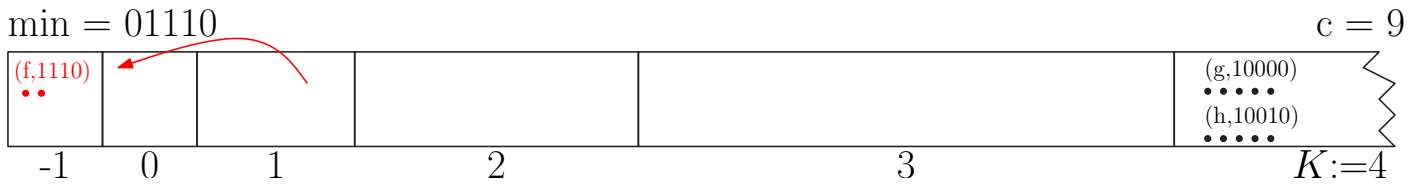
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

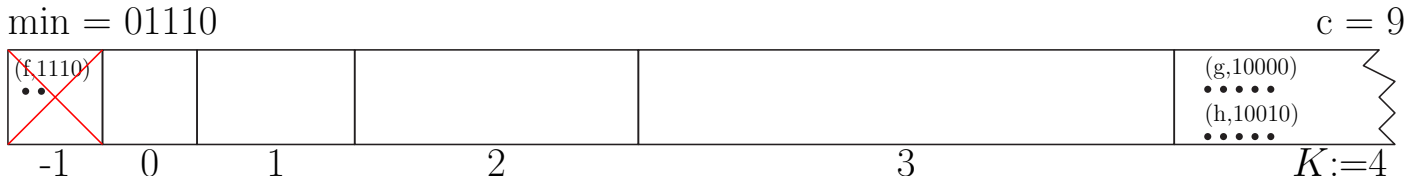
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

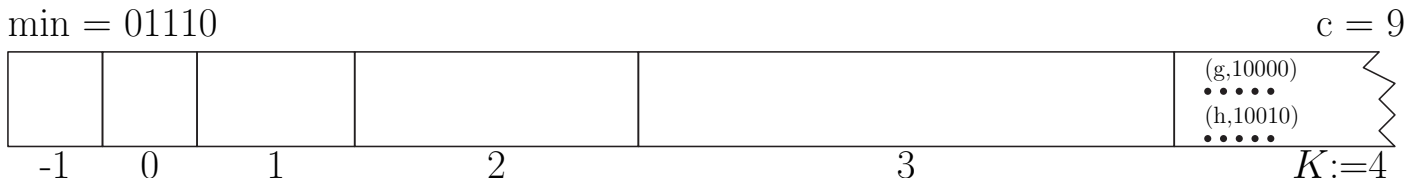
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

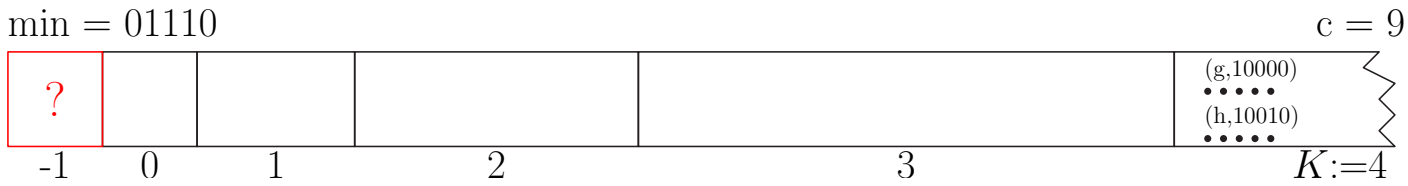
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

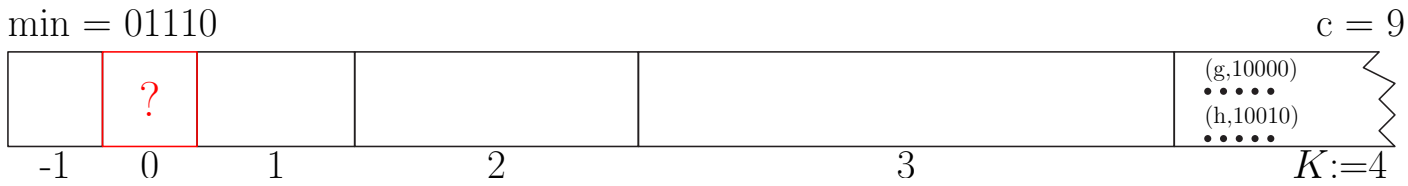
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

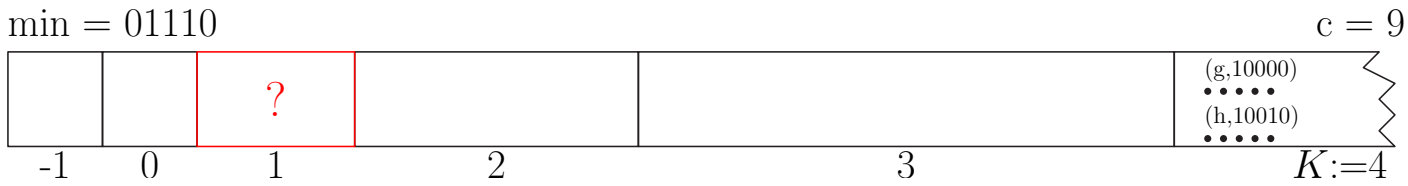
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

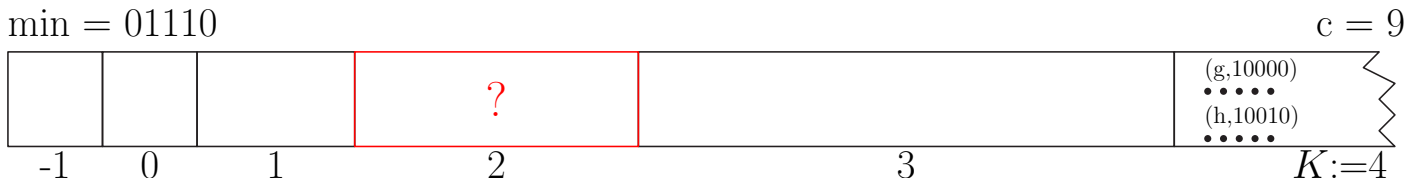
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

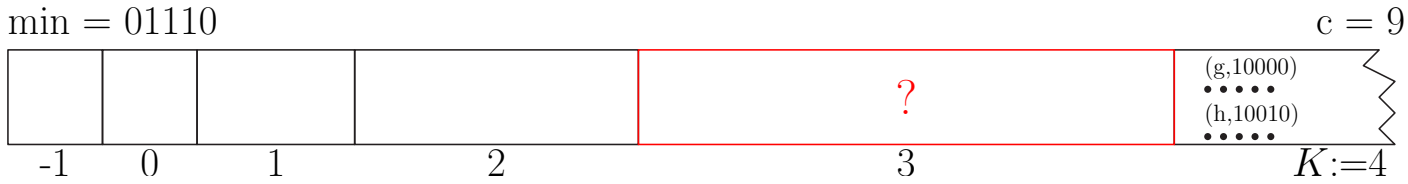
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

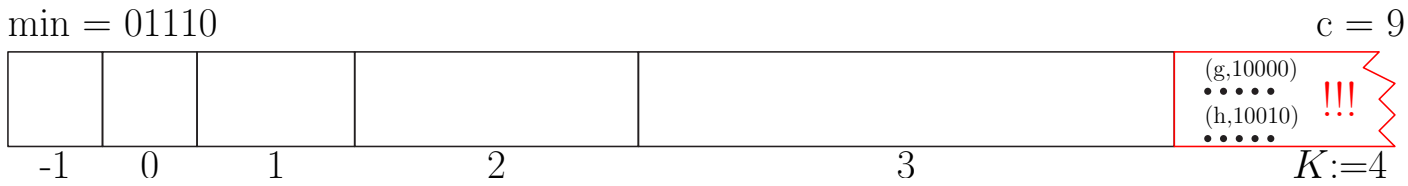
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

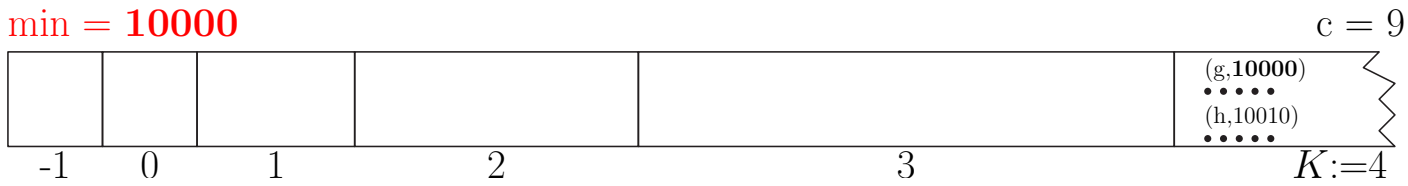
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

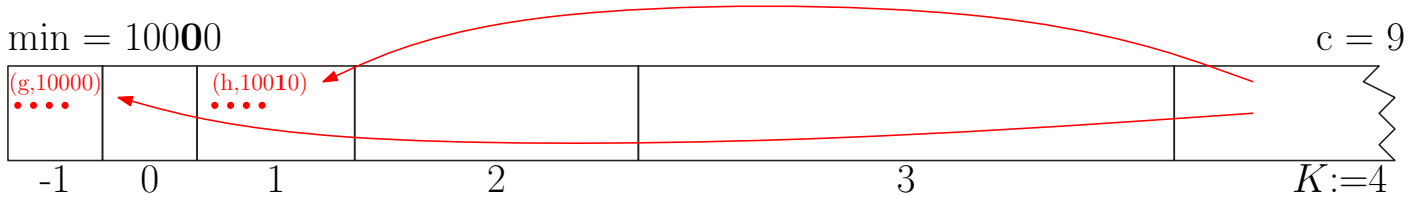
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

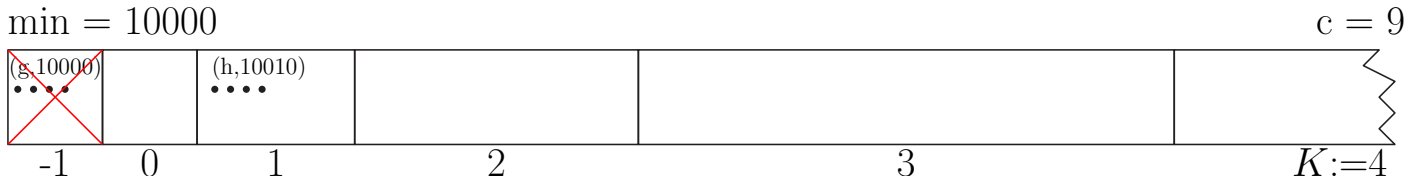
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

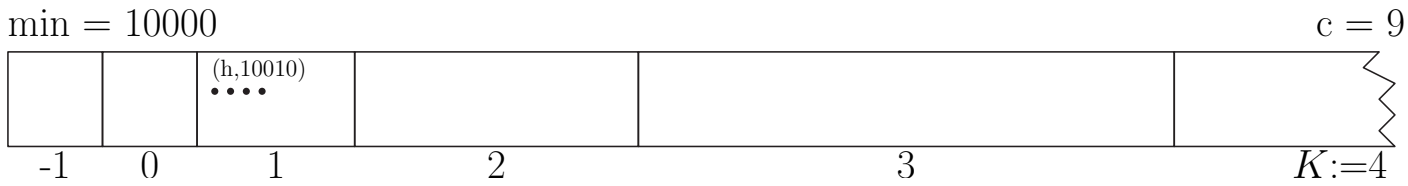
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

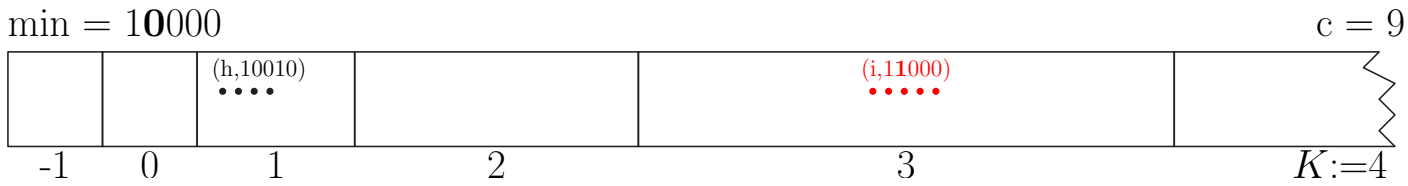
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

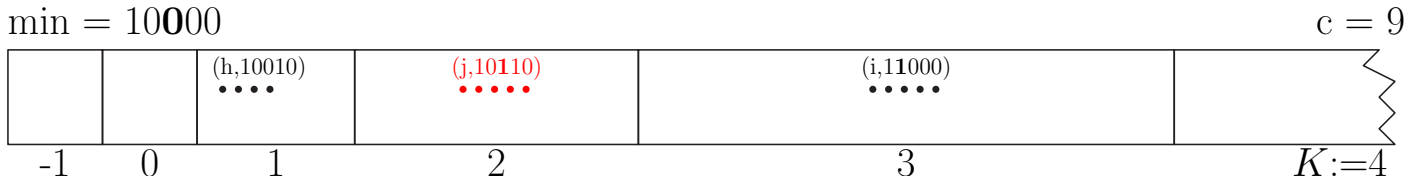
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

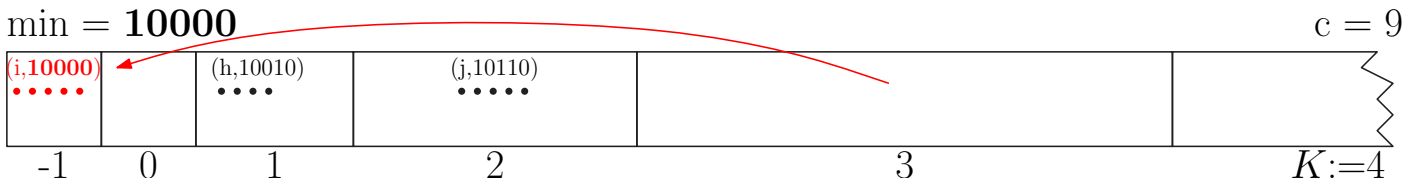
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

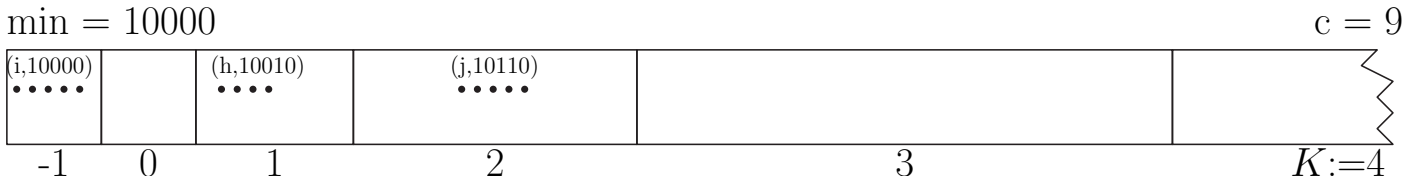
- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Beispiel (fortgesetzt):

- `insert(f,14)`
- `deleteMin()`
- `deleteMin()`
- `insert(g,20)`
- `deleteMin()`
- `insert(h,18)`
- `deleteMin()`
- `decreaseKey(g,16)`
- `deleteMin()`
- `deleteMin()`
- `insert(i,24)`
- `insert(j,22)`
- `decreaseKey(i,16)`



Radix Heaps

Warum komplizierte Formel $\min(\text{msd}(\mathit{min}, \mathit{key}), K)$ *?*

- viel anschaulicher wäre doch

$$i = \lceil \log(\mathit{key} - \mathit{min}) \rceil$$

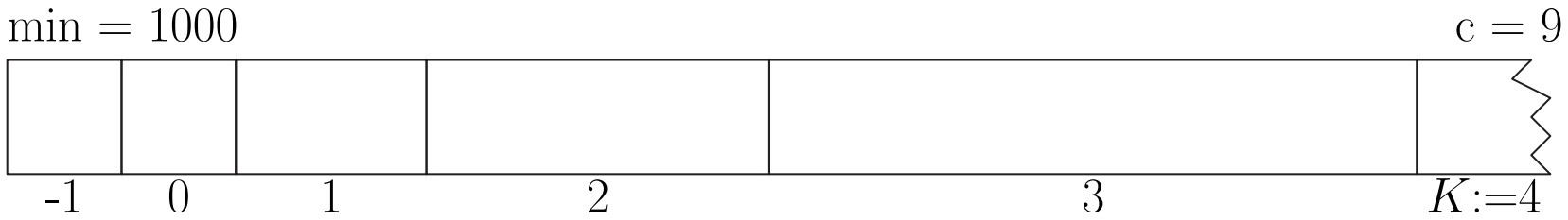
- okay, Anzahl Verschiebungen erhöht sich nicht ...
- ... aber, jede Änderung von *min* ändert potentiell alle Buckets
  $\rightarrow$ schlechtere Laufzeit!

Beispiel

$\mathit{min} := \texttt{01000}\ (08) \xrightarrow{\text{deleteMin}} \mathit{min} := \texttt{01010}\ (10)$

	Bucket $B[\cdot]$				
min	0	1	2	3	4
1000 (08)	8	9..10	11..12	13..16	17..8+C
1010 (10)	10	11..12	13..14	15..18	19..10+C

Radix Heaps

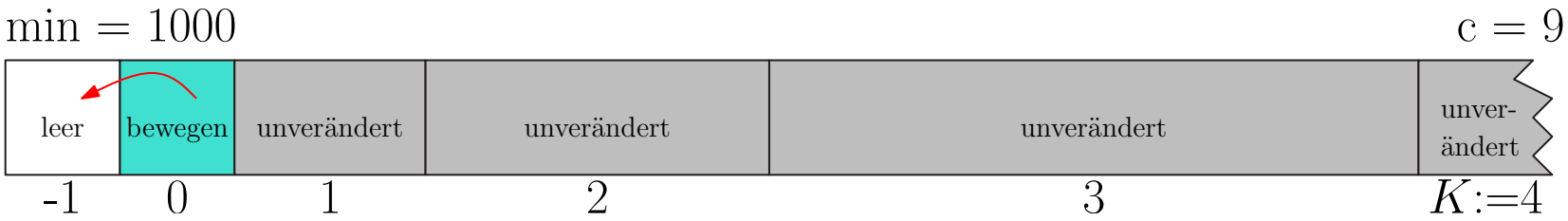


Änderung von min $\longrightarrow$ *Umverteilung eines Bucket genügt*

min $:=$ 01000 (08)

	Bucket $B[\cdot]$					
min	-1	0	1	2	3	4
01000 (08)	8	9	10..11	12..15	-	≥ 16

Radix Heaps

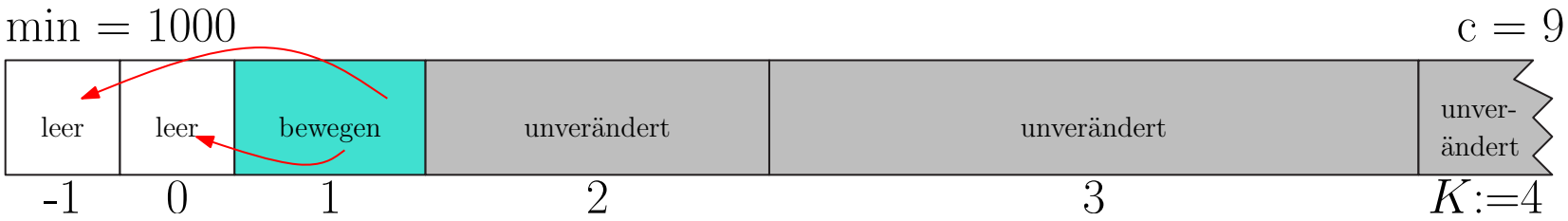


Änderung von min $\longrightarrow$ *Umverteilung eines Bucket genügt*

min $:=$ 01000 (08) $\overset{\text{deleteMin}}{\longrightarrow}$ *min* $:=$ 01001 (09)

	Bucket $B[\cdot]$					
min	-1	0	1	2	3	4
01000 (08)	8	9	10..11	12..15	-	≥ 16
01001 (09)	9	-	10..11	12..15	-	≥ 16

Radix Heaps

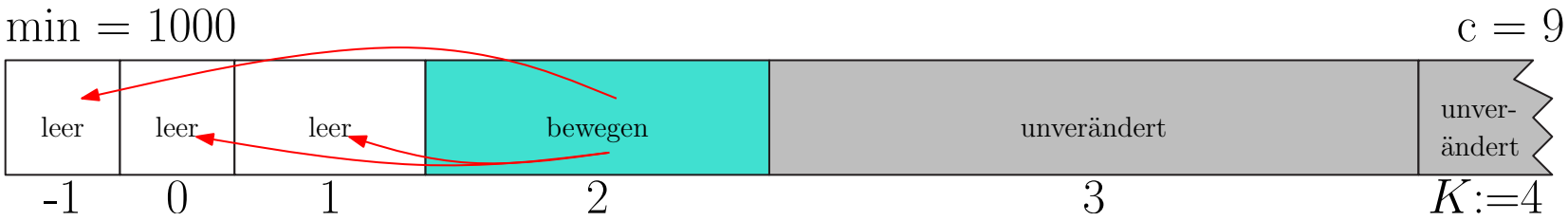


Änderung von min $\longrightarrow$ *Umverteilung eines Bucket genügt*

min := 01000 (08) $\overset{\text{deleteMin}}{\longrightarrow}$ *min* := 0101∗ (10..11)

	Bucket $B[\cdot]$					
min	-1	0	1	2	3	4
01000 (08)	8	9	10..11	12..15	-	≥ 16
01010 (10)	10	11	-	12..15	-	≥ 16
01011 (11)	11	-	-	12..15	-	≥ 16

Radix Heaps

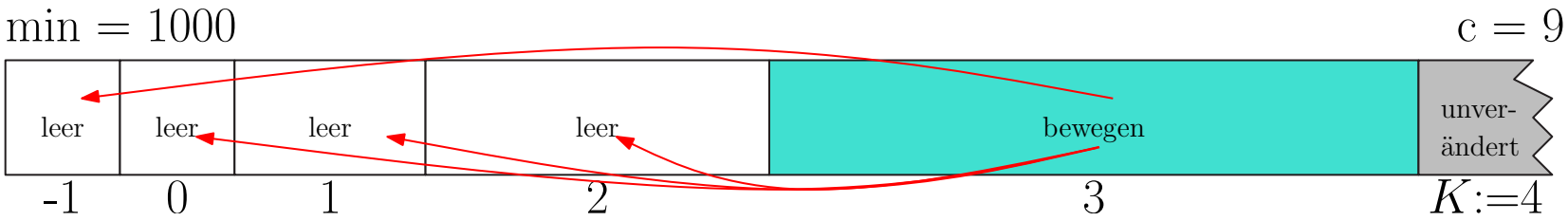


Änderung von min $\longrightarrow$ *Umverteilung eines Bucket genügt*

min $:= 01000\ (08) \xrightarrow{\text{deleteMin}}$ *min* $:= 011**\ (12..15)$

	Bucket $B[\cdot]$					
min	-1	0	1	2	3	4
01000 (08)	8	9	10..11	12..15	-	≥ 16
01100 (12)	12	13	14..15	-	-	≥ 16
01101 (13)	13	-	14..15	-	-	≥ 16
01110 (14)	14	15	-	-	-	≥ 16
01111 (15)	15	-	-	-	-	≥ 16

Radix Heaps

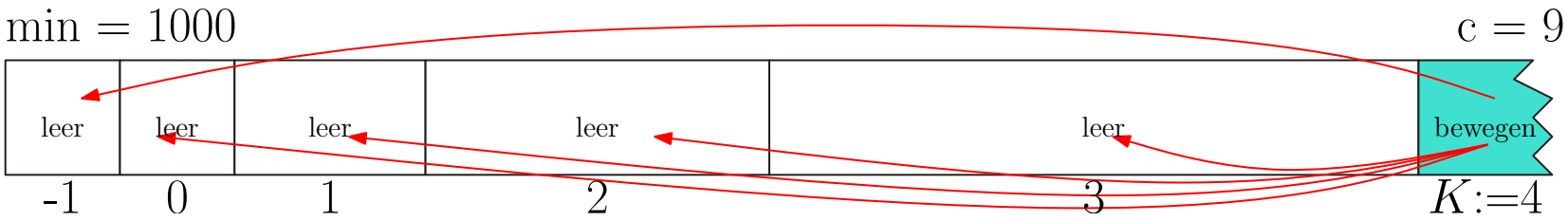


Änderung von min $\longrightarrow$ *Umverteilung eines Bucket genügt*

min $:= 01000\ (08) \xrightarrow{\text{deleteMin}}$ *min* $:= -$, *Bucket 3 ist leer*

	Bucket $B[\cdot]$					
min	-1	0	1	2	3	4
01000 (08)	8	9	10..11	12..15	-	≥ 16

Radix Heaps



Änderung von min $\longrightarrow$ *Umverteilung eines Bucket genügt*

min := 01000 (08) $\xrightarrow{\text{deleteMin}}$ *min* := 1 ∗ ∗ ∗ ∗ (16..)

	Bucket $B[\cdot]$					
min	-1	0	1	2	3	4
01000 (08)	8	9	10..11	12..15	-	$\geq$ 16
10000 (16)	16	17	-	-	-	-
10001 (17)	17	-	-	-	-	-

Ende!

Feierabend!