Informatik

Dynamisches Hashing:

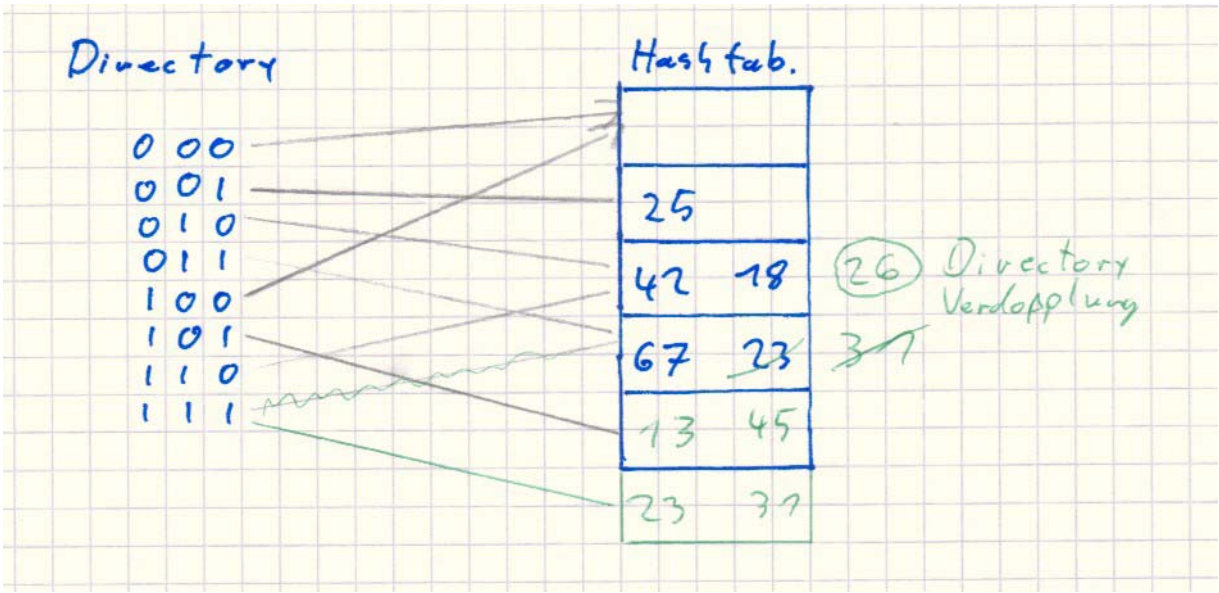
i) extendible hashing

hash (x) = x mod 4 →
b stelliges Bitmuster
letzte i Stellen werden zur Zuordnung verwendet

Directory	Hastable
00	
01	25 13 45
10	42 18
11	67 23

Einfügen von:
25, 13, 42, 45, 67, 18, 23, 31 hash (x) = x

Erweiterte Hashtabelle:



(Bild 3.1)

ii) linear hashing

Hash-Funktion, Zuordnung wie für i)

	25 13	42 18	67 23
	45		

Neuer Eintrag in Hashtabelle, sobald Block überfüllt
Splits erfolgen von links nach rechts

	25 13	42 18	67 23		13 45
	~~45~~		31		

Text: M. Wilke
Zeichnungen: M Löschinger

Sortieren:

Merge-sort

7 4 2 6 3 1 9 5

47	26	13	59

2 4 6 7	1 3 5 9

1 2 3 4 5 6 7 9

Selection sort (Sortieren durch Auswahl)

7 4 2 6 3 1 9 5	Quelle
1	Ziel

Schritt 1: Bestimme Minimum → 1
Einfügen in Ziel
Löschen in Quelle
(Effizienz vs. Stabilität)

Schritt solange wiederholen, bis Quelle verarbeitet ist.

Anzahl der Vergleiche = O(n^2) = ((n-1)n)/2
Anzahl der Vertauschungen = O(n) = n-1
Anzahl der Kopieroperationen (löschen) = O(n^2) = (n^2)/4
Anzahl der Einfügeoperationen = O(n) = n

$7\ 4\ 2\ 6\ 3\ 1\ 9^1\ 5\ 9^2$
$1 \mid 4\ 2\ 6\ 3\ 7\ 9^1\ 5\ 9^2$
$1\ 2 \mid 4\ 6\ 3\ 7\ 9^1\ 5\ 9^2$
| noch zu sortieren

Instertion sort

$7 \mid 4\ 2\ 6\ 3\ 1\ 9^1\ 5\ 9^2$
$4\ 7 \mid 2\ 6\ 3\ 1\ 9^1\ 5\ 9^2$
$2\ 4\ 7 \mid 6\ 3\ 1\ 9^1\ 5\ 9^2$

Geeignet für vorsortierte oder kleine Folgen (Listen)

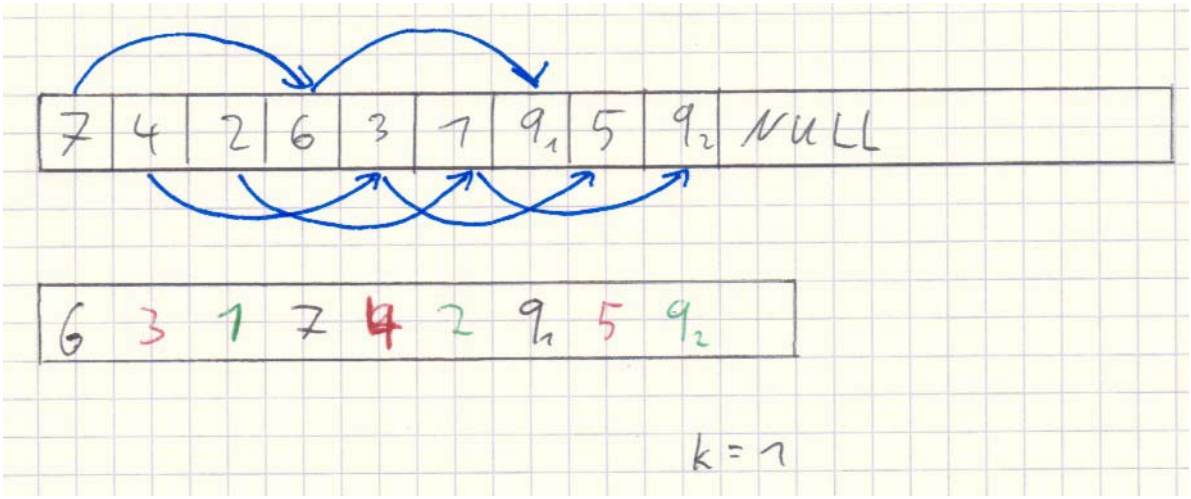
usw.

$1\ 2\ 3\ 4\ 6\ 7 \mid 9^1\ 5\ 9^2$
$1\ 2\ 3\ 4\ 6\ 7\ 9^1 \mid 5\ 9^2$
$1\ 2\ 3\ 4\ 5\ 6\ 7\ 9^1 \mid 9^2$

Shell sort

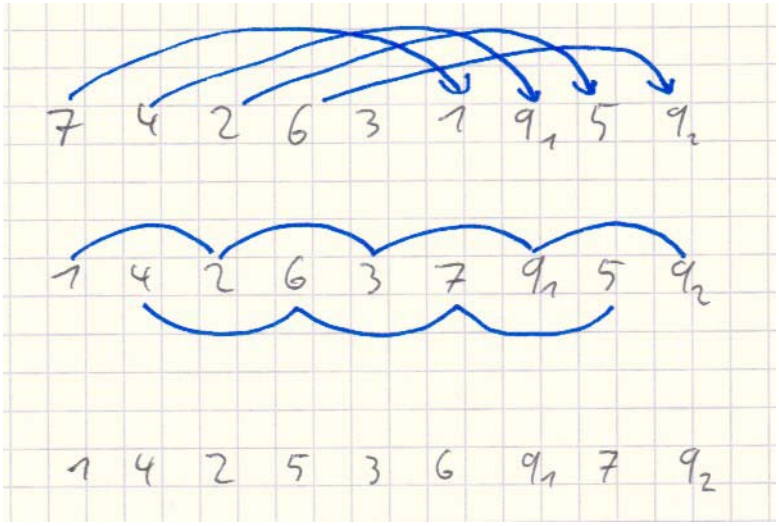
Schrittweite h = (2^n)-1 z.B. h = 3

Text: M. Wilke
Zeichnungen: M Löschinger



(Bild 3.2)

Sehr gut in Kombination mit Insertion Sort hinterher.

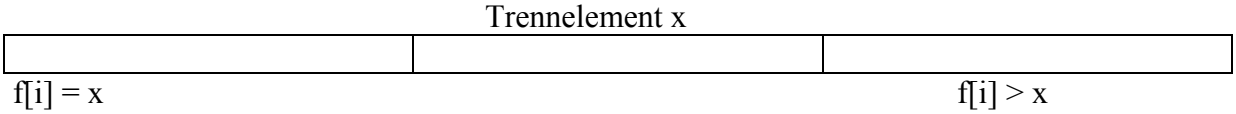


(Bild 3.3)

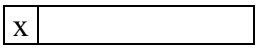
Text: M. Wilke
Zeichnungen: M Löschinger

Quicksort

Prinzip: Teile und herrsche



schlecht



Algorithmus:

1) Unterteile die Folge derart, dass ein Trennelement x an der richtigen Position i° steht und folgendes gilt.
   $f(i) < x$ für $i < i°$
   $f(i) > x$ für $i > i°$
2) Quicksort im Bereich istart, ..., i 0 – 1
3) Quicksort im Bereich i 0 + 1, ..., iend

Trennelement x muss mittig in der Folge vorkommen. („7“ im bisherigem Bsp.)

Text: M. Wilke
Zeichnungen: M Löschinger