Suchen und Sortieren

- Suchen
  - Lineare Suche
  - Binäre Suche
- Sortieren
  - Insertion sort
  - Merge sort

Holger Dell
Folien adaptiert von Philip Bille

Suchen und Sortieren

- Suchen
  - Lineare Suche
  - Binäre Suche
- Sortieren
  - Insertion sort
  - Merge sort

Suchen

- Suchen.
  Gegeben ein sortiertes Feld A und eine Zahl x, finde heraus, ob x im Feld vorkommt.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	4	7	12	16	18	25	28	31	33	36	42	45	47	50

Lineare Suche

- Lineare Suche. Teste für jeden Eintrag, ob er x ist.
- Zeit?
- Frage. Können wir ausnutzen, dass das Feld sortiert ist?

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	4	7	12	16	18	25	28	31	33	36	42	45	47	50

Binäre Suche

- Binäre Suche. Vergleiche den mittleren Eintrag A[m] mit x.
  - wenn A[m] = x, gebe true zurück und fertig.
  - wenn A[m] < x, suche rekursiv in der rechten Hälfte weiter.
  - wenn A[m] > x, suche rekursiv in der linken Hälfte weiter.
- Wenn das Feld Größe ≤ 0 hat, gebe false zurück und fertig.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	4	7	12	16	18	25	28	31	33	36	42	45	47	50

33?

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	4	7	12	16	18	25	28	31	33	36	42	45	47	50

8?

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	4	7	12	16	18	25	28	31	33	36	42	45	47	50

Binäre Suche

```
BinarySearch(A,i,j,x)
    if j < i return false
    m = ⌊(i+j)/2⌋
    if A[m] = x return true
    elseif A[m] < x return BinarySearch(A,m+1,j,x)
    else return BinarySearch(A,i,m-1,x)     // A[m] > x
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	4	7	12	16	18	25	28	31	33	36	42	45	47	50

- Zeit?
- Analyse 1. Analog zum rekursiven Hügelalgorithmus.
  - Ein rekursiver Aufruf braucht konstante Zeit.
  - Jeder rekursive Aufruf halbiert die Größe des Felds.
    Wir stoppen, wenn die Größe ≤ 0 ist.
  - ⇒ Laufzeit ist O(log n)

Binäre Suche

- Analyse 2. Sei T(n) die Laufzeit von binärer Suche.
  - Löse die Rekursionsgleichung von T(n).

$$T(n) = \begin{cases} T(n/2) + c & \text{if } n > 1 \\ d & \text{otherwise} \end{cases}$$

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + c \\ &= T\left(\frac{n}{4}\right) + c + c \\ &= T\left(\frac{n}{8}\right) + c + c + c \\ &\vdots \\ &= T\left(\frac{n}{2^k}\right) + ck \\ &\vdots \\ &= T\left(\frac{n}{2^{\log_2 n}}\right) + c \log_2 n \\ &= T(1) + c \log_2 n \\ &= d + c \log_2 n \\ &= O(\log n) \end{aligned}$$

Suchen

- Wir können suchen in Zeit
  - O(n) mit linearer Suche.
  - O(log n) mit binärer Suche.

Suchen und Sortieren

- Suchen
  - Lineare Suche
  - Binäre Suche
- Sortieren
  - Insertion sort
  - Merge sort

Sortieren

- Sortieren. Gegeben Feld A[0..n-1], gib Feld B[0..n-1] zurück mit denselben Einträgen wie A, aber in sortierter Reihenfolge.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
33	4	25	28	45	18	7	12	36	1	47	42	50	16	31

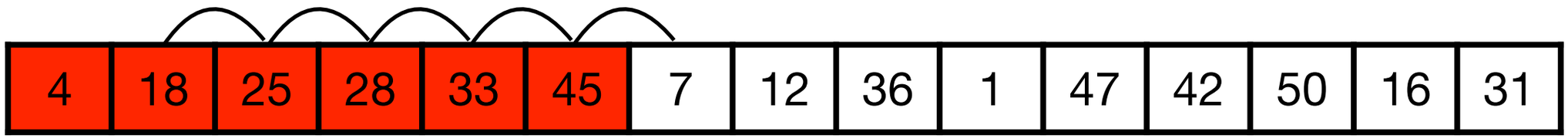
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	4	7	12	16	18	25	28	31	33	36	42	45	47	50

Anwendungen

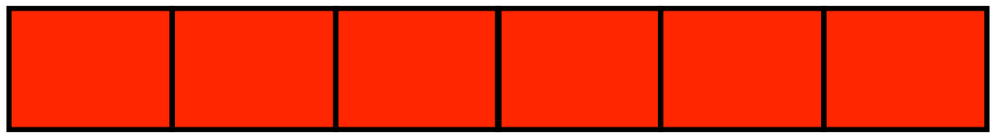
- Offensichtliche.
  - Sortiere Liste von Namen, sortiere Google Suchergebnisse nach PageRank, zeige Instagramfeed in chronologischer Reihenfolge.
- Nicht offensichtliche.
  - Datenkompression, Computergraphik, Bioinformatik, Empfehlungssysteme.
- Einfache Probleme für sortierte Daten.
  - Suchen, finde Median, finde Duplikate, finde kleinste Differenz, finde Ausreißer.

Insertion Sort

- Insertion sort. Starte mit unsortiertem Feld A.
- Schreite von links nach rechts in n Runden.
- Runde i:
  - Teilfeld A[0..i-1] ist sortiert.
  - Füge A[i] in A[0..i-1] ein, so dass A[0..i] sortiert ist.



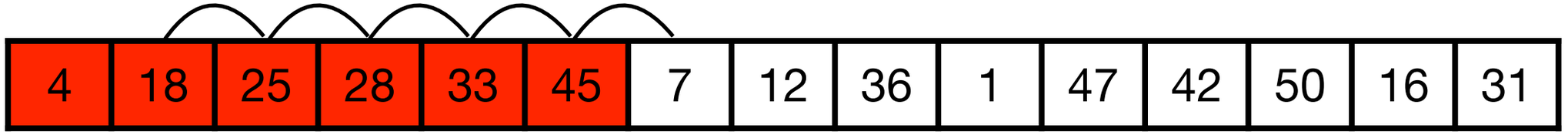
4 5 20 25 33 45



33 4 25 5 20 45

Insertion Sort

```
InsertionSort(A, n)
   for i = 1 to n-1
      j = i
      while j > 0 and A[j-1] > A[j]
         swap A[j] and A[j-1]
         j = j - 1
```



- Zeit?
  - Um A[i] einzufügen, brauchen wir Zeit c·i für eine Konstante c.
  - $\Rightarrow$ Gesamtzeit T(n):

$$T(n) = \sum_{i=1}^{n-1} ci = c \sum_{i=1}^{n-1} i = \frac{cn(n-1)}{2} = O(n^2)$$

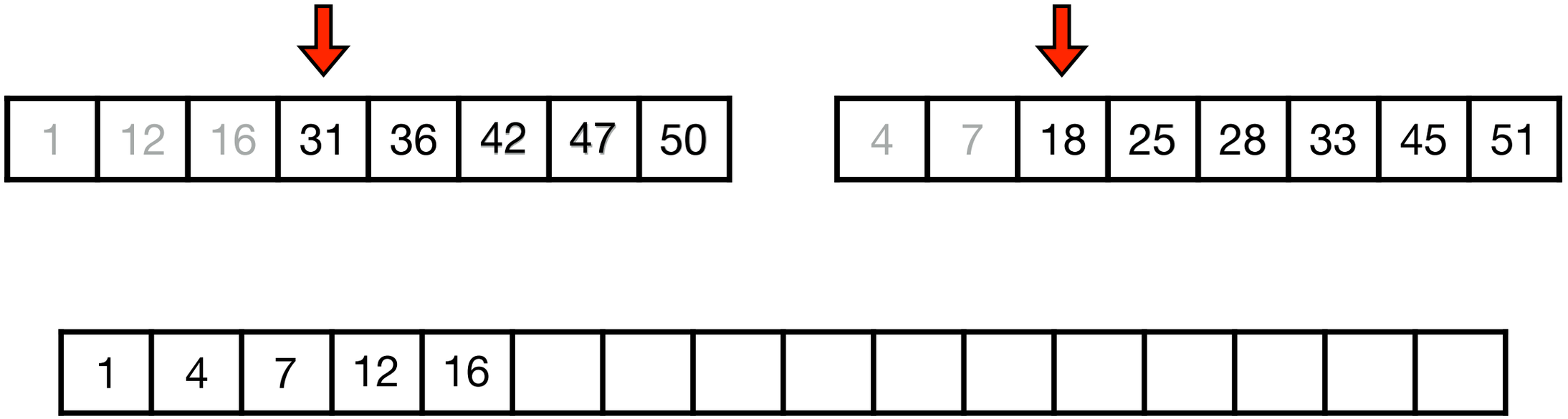
- Frage. Schneller sortieren?

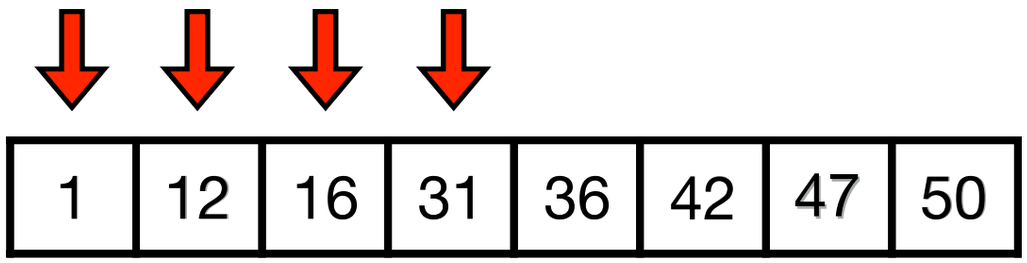
Merge sort

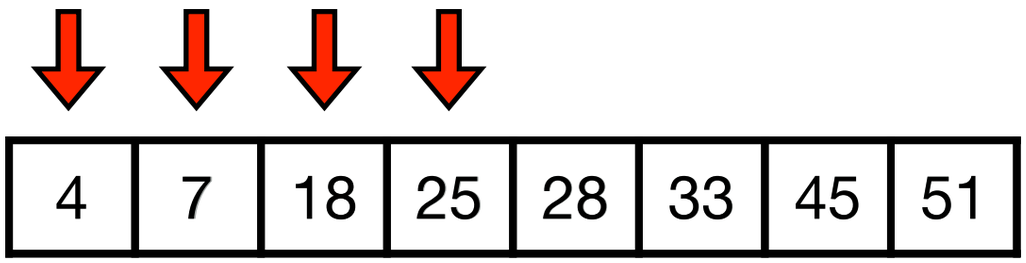
- Merge sort.
  - Idee. Rekursiv sortieren durch Verflechten sortierter Teilfelder.

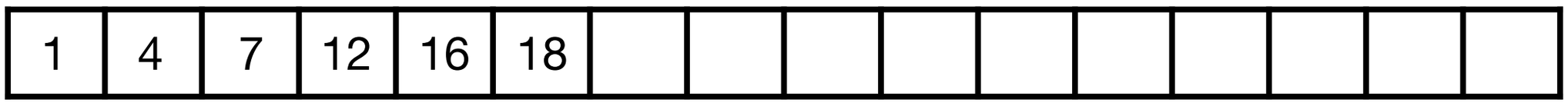
Verflechten (*Merge*)

- Ziel. Verflechte zwei sortierte Felder in ein einziges sortiertes Feld.
- Idee.
  - Durchlaufe beide Felder von links nach rechts. In jedem Schritt:
    - Füge das kleinste der beiden Einträge in das neue Feld ein.
    - Im Feld mit dem kleinsten Eintrag, geh einen Schritt nach rechts.
    - Wiederhole, bis beide Eingabefelder leer sind.



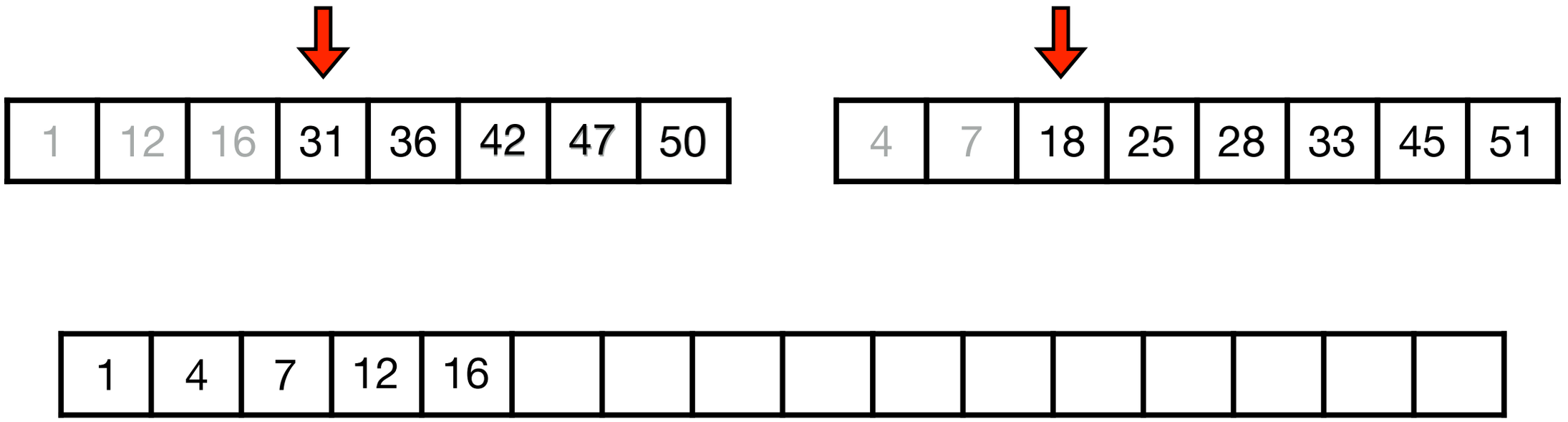

1
12
16
31
36
42
47
50


4
7
18
25
28
33
45
51


1
4
7
12
16
18

Verflechten (*Merge*)

- Zeit. Verflechten von zwei Feldern A_1 und A_2?
  - Jeder Schritt braucht Zeit O(1).
  - In jedem Schritt gehen wir in einem Feld eins weiter.
  - ⇒ O($|A_1| + |A_2|$) Zeit.



Merge Sort

- Merge sort.
- Wenn |A| ≤ 1, gib A zurück.
- Sonst:
  - Teile A in zwei Hälften.
  - Sortiere jede Hälfte rekursiv.
  - Verflechte die beiden Hälften.

16	31	1	36	47	50	42	12	7	4	51	28	45	25	18	33
1	4	7	12	16	18	25	28	31	33	36	42	45	47	50	51

16	31	1	36	47	50	42	12
1	12	16	31	36	42	47	50

7	4	51	28	45	25	18	33
4	7	18	25	28	33	45	51

16	31	1	36	47	50	42	12	7	4	51	28	45	25	18	33
1	4	7	12	16	18	25	28	31	33	36	42	45	47	50	51

16	31	1	36	47	50	42	12
1	12	16	31	36	42	47	50

7	4	51	28	45	25	18	33
4	7	18	25	28	33	45	51

16	31	1	36
1	16	31	36

47	50	42	12
12	42	47	50

7	4	51	28
4	7	28	51

45	25	18	33
18	25	33	45

16	31	1	36	47	50	42	12
16	31	1	36	47	50	12	42

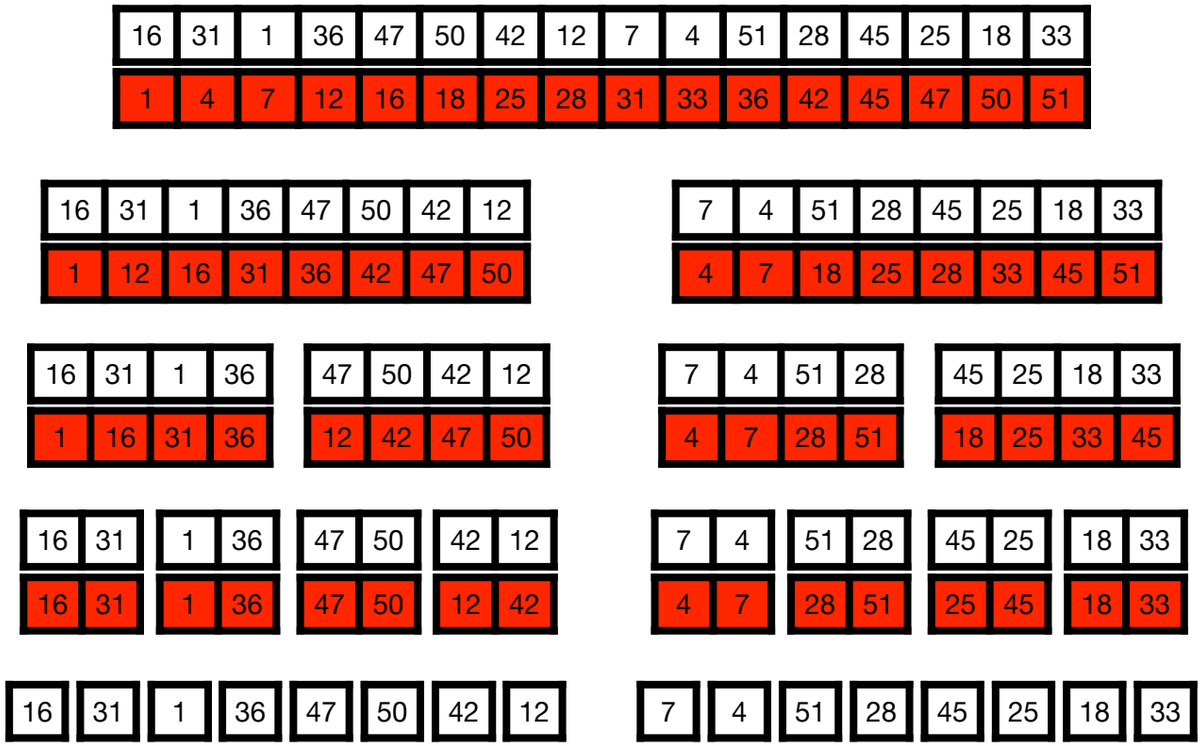
7	4	51	28	45	25	18	33
4	7	28	51	25	45	18	33

16	31	1	36	47	50	42	12

7	4	51	28	45	25	18	33

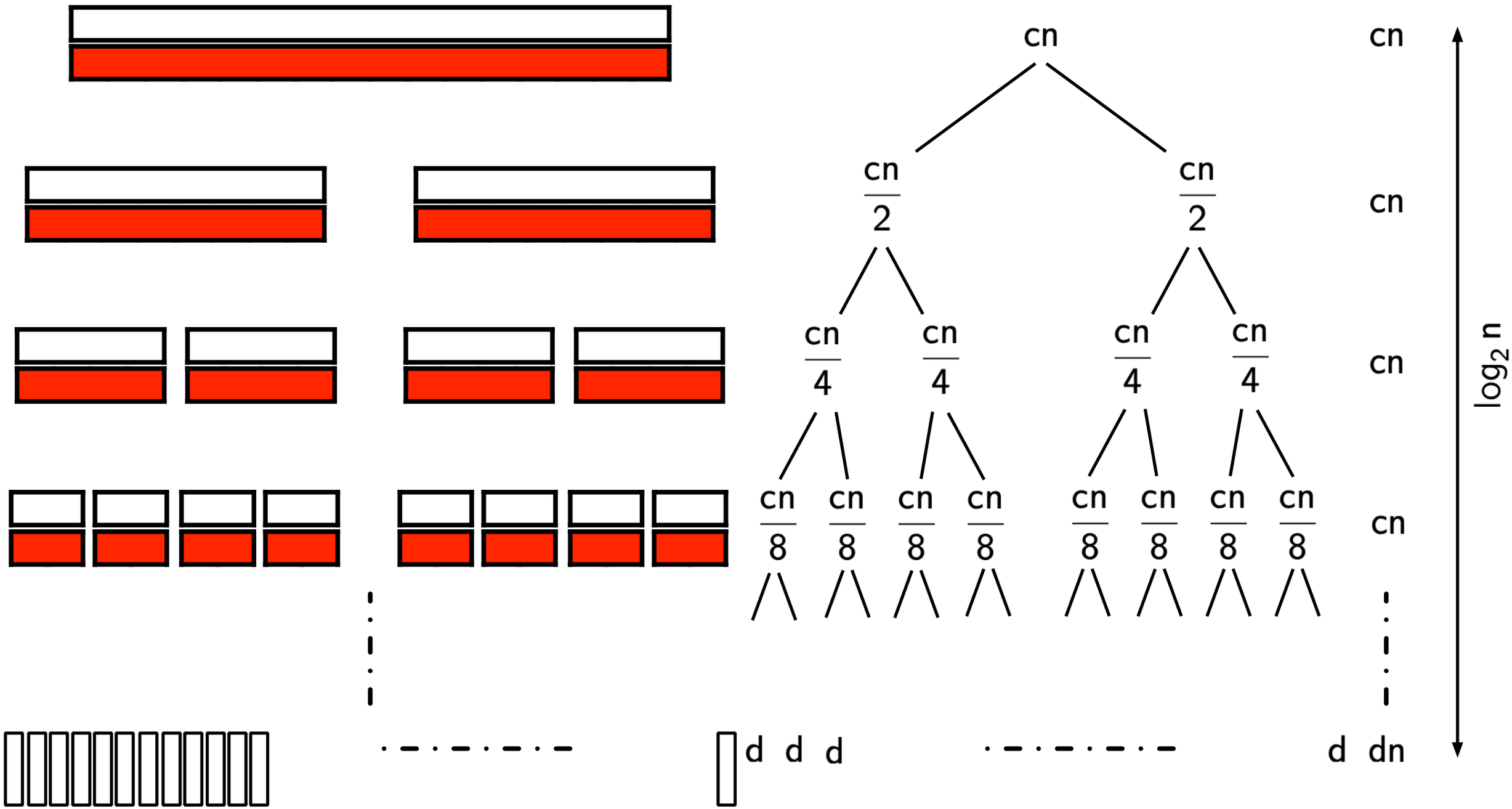
Merge Sort

```
MergeSort(A,i,j)
    if i < j
        m =⌊(i+j)/2⌋
        MergeSort(A, i, m)
        MergeSort(A, m+1, j)
        Merge(A, i, m, j)
```



- Zeit?
- Wir schauen auf den Rekursionsbaum.

Merge Sort



$$T(n) = cn\log_2 n + dn = O(n\log n)$$

Sortieren

- Wir können sortieren in Zeit
  - $O(n^2)$ mit insertion sort.
  - O(n log n) mit merge sort.

Teile und Herrsche (*Divide and Conquer*)

- Merge sort ist ein divide and conquer Algorithmus.
- Algorithmische Entwurfstechnik.
  - Teile das Problem in Teilprobleme.
  - Herrsche: Löse die Teilprobleme rekursiv.
  - Kombiniere Lösungen für Teilprobleme zu einer Lösung für das Problem.
- Merge sort.
  - Teile das Feld in zwei Hälften.
  - Herrsche: Sortiere jede Hälfte.
  - Kombiniere: Verflechte die sortierten Hälften.

Suchen und Sortieren

- Suchen
  - Lineare Suche
  - Binäre Suche
- Sortieren
  - Insertion sort
  - Merge sort