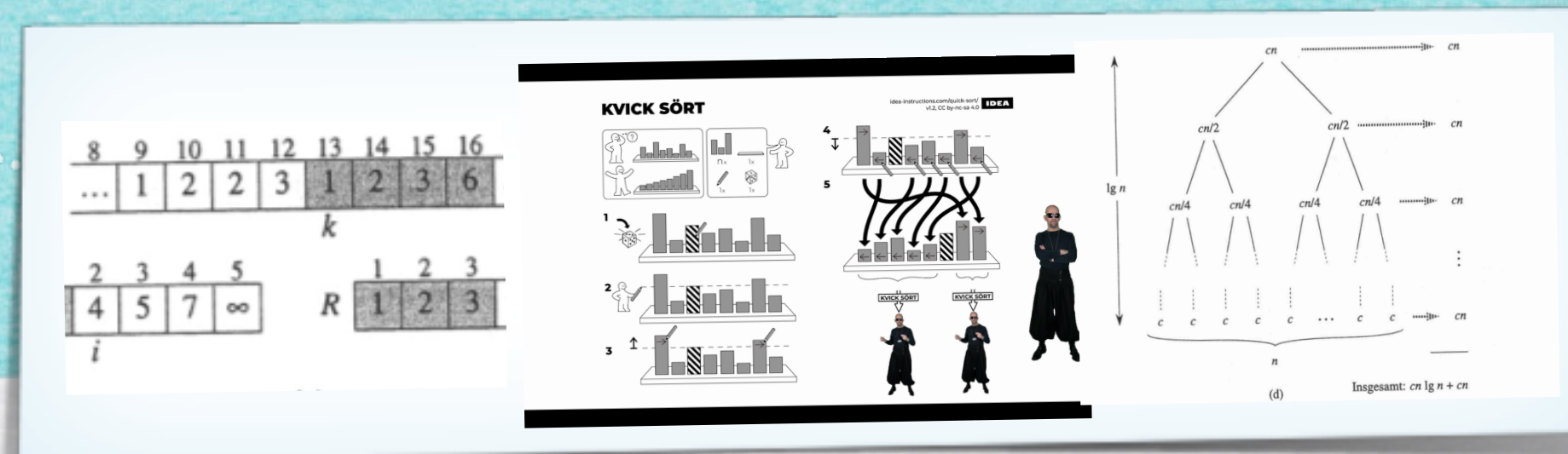
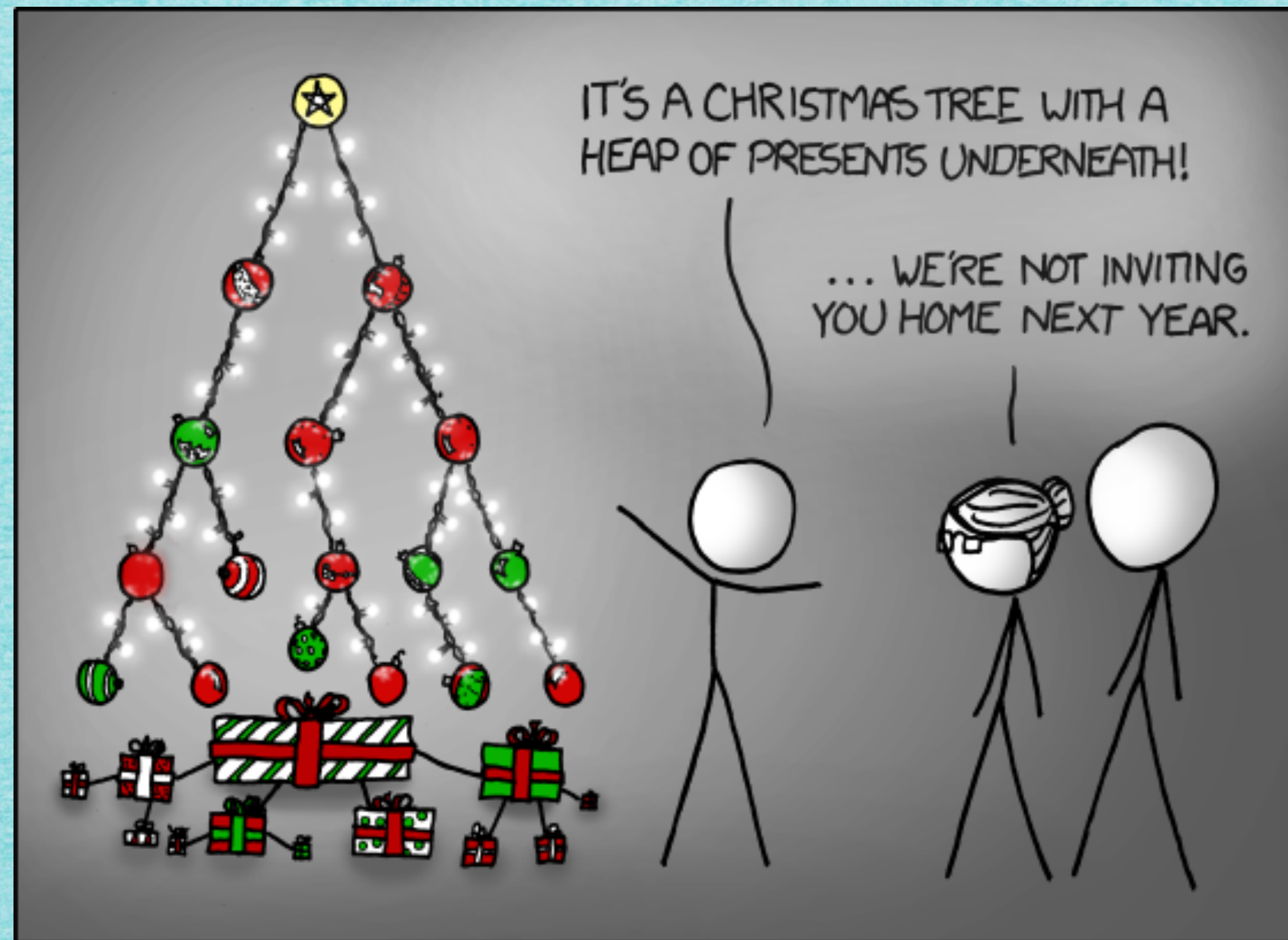




Kapitel 5: Sortieren

Algorithmen und Datenstrukturen
WS 2025/26

Prof. Dr. Sándor Fekete



Wer ist eigentlich dieser knut?

Knut hat in Schweden eine fast 1.000 Jahre alte Tradition. Der **13.1.**, der St.-Knuts-Tag, hat seinen Namen von Knut IV., einem König, der von Weihnachten nicht genug bekommen konnte. Also hat er das **Fest auf 20 Tage verlängert**. Seitdem endet es jedes Jahr mit dem Knut-Tag, an dem wir **abschmücken und Platz für Neues schaffen**.



Donald Knuth



Knuth in 2005

Born

Donald Ervin Knuth

January 10, 1938 (age 87)

Milwaukee, Wisconsin, U.S.



Donald Knuth



Knuth in 2005

Born

Donald Ervin Knuth
January 10, 1938 (age 87)
[Milwaukee, Wisconsin, U.S.](#)

THE CLASSIC WORK
NEWLY UPDATED AND REVISED

The Art of Computer Programming

VOLUME 3

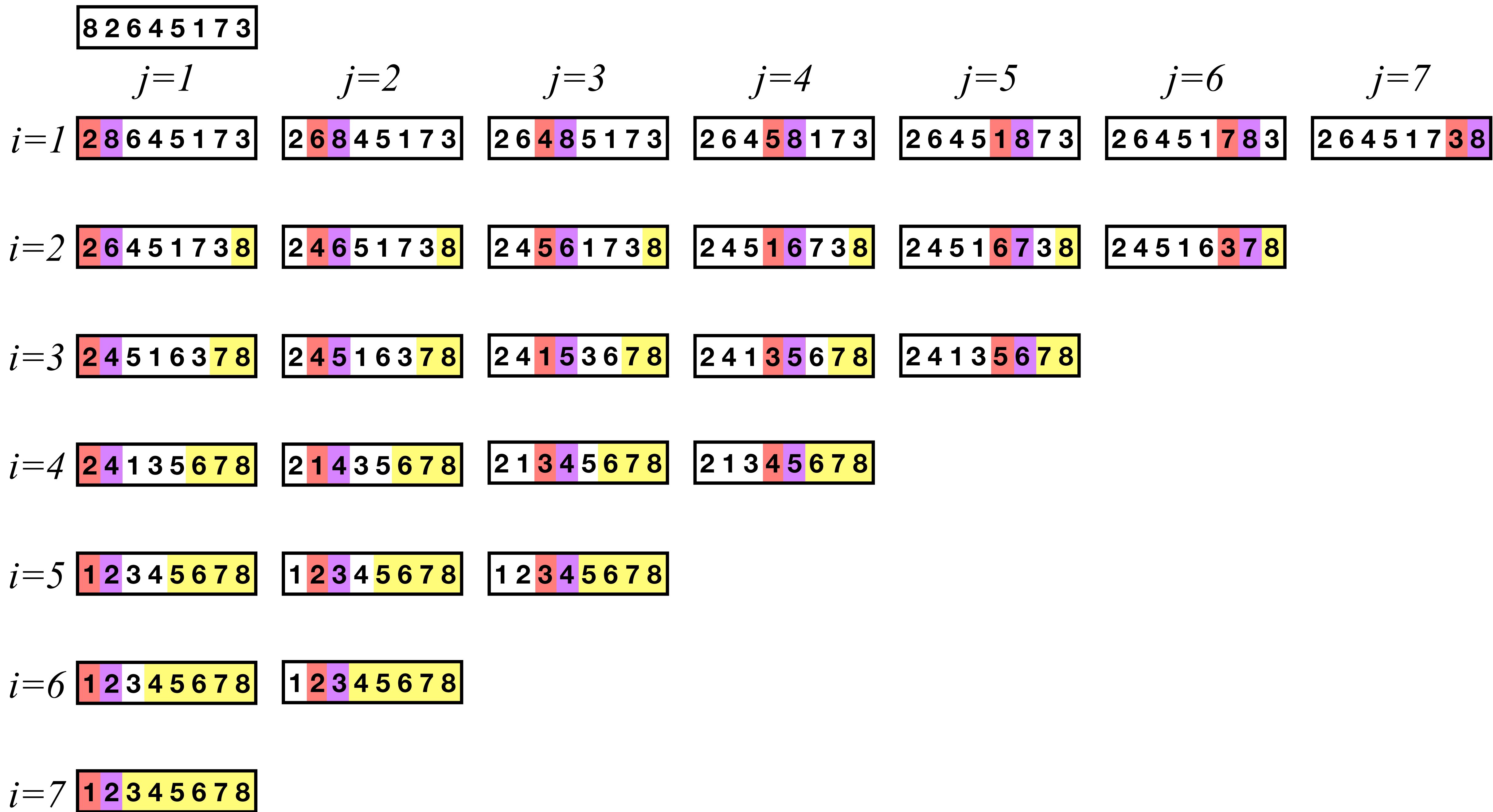
Sorting and Searching
Second Edition

DONALD E. KNUTH

5.0 Aufgabenstellung

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]
8	2	6	4	5	1	7	3

Bubblesort



Selectionsort

8 2 6 4 5 1 7 3

$j=2$

$j=3$

$j=4$

$j=5$

$j=6$

$j=7$

$j=8$

$i=1$

8 2 6 4 5 1 7 3

8 2 6 4 5 1 7 3

8 2 6 4 5 1 7 3

8 2 6 4 5 1 7 3

8 2 6 4 5 1 7 3

8 2 6 4 5 1 7 3

3 2 6 4 5 1 7 8

$i=2$

3 2 6 4 5 1 7 8

3 2 6 4 5 1 7 8

3 2 6 4 5 1 7 8

3 2 6 4 5 1 7 8

3 2 6 4 5 1 7 8

3 2 6 4 5 1 7 8

$i=3$

3 2 6 4 5 1 7 8

3 2 6 4 5 1 7 8

3 2 6 4 5 1 7 8

3 2 6 4 5 1 7 8

3 2 1 4 5 6 7 8

$i=4$

3 2 1 4 5 6 7 8

3 2 1 4 5 6 7 8

3 2 1 4 5 6 7 8

3 2 1 4 5 6 7 8

$i=5$

3 2 1 4 5 6 7 8

3 2 1 4 5 6 7 8

3 2 1 4 5 6 7 8

$i=6$

3 2 1 4 5 6 7 8

1 2 3 4 5 6 7 8

$i=7$

1 2 3 4 5 6 7 8

Insertionsort

8 2 6 4 5 1 7 3

$i=2$

8 2 6 4 5 1 7 3

2 8 6 4 5 1 7 3

$i=3$

2 8 6 4 5 1 7 3

2 6 8 4 5 1 7 3

$i=4$

2 6 8 4 5 1 7 3

2 6 4 8 5 1 7 3

2 4 6 8 5 1 7 3

$i=5$

2 6 8 4 5 1 7 3

2 4 6 5 8 1 7 3

2 4 5 6 8 1 7 3

$i=6$

2 4 5 6 8 1 7 3

2 4 5 6 1 8 7 3

2 4 5 1 6 8 7 3

2 4 1 5 6 8 7 3

2 1 4 5 6 8 7 3

1 2 4 5 6 8 7 3

$i=7$

1 2 4 5 6 8 7 3

1 2 4 5 6 7 8 3

$i=8$

1 2 4 5 6 7 8 3

1 2 4 5 6 7 3 8

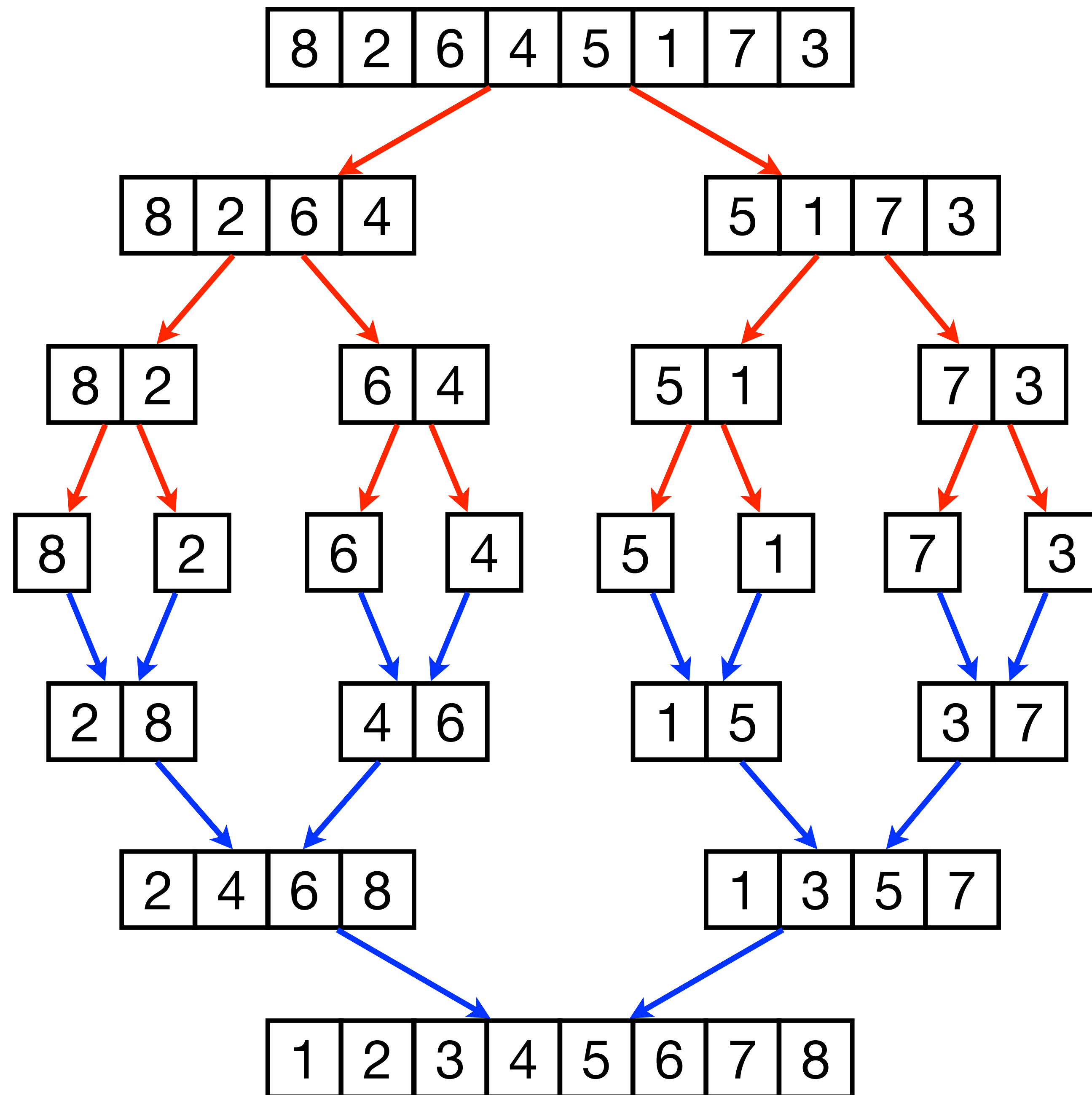
1 2 4 5 6 3 7 8

1 2 4 5 3 6 7 8

1 2 4 3 5 6 7 8

1 2 3 4 5 6 7 8

Mergesort



5.1 Mergesort



5.1.2 Algorithmische Beschreibung

Algorithmus 5.1

INPUT: Subarray von $A=[1,\dots,n]$,
der bei Index p beginnt und bei Index r endet, d.h. $A[p,\dots,r]$

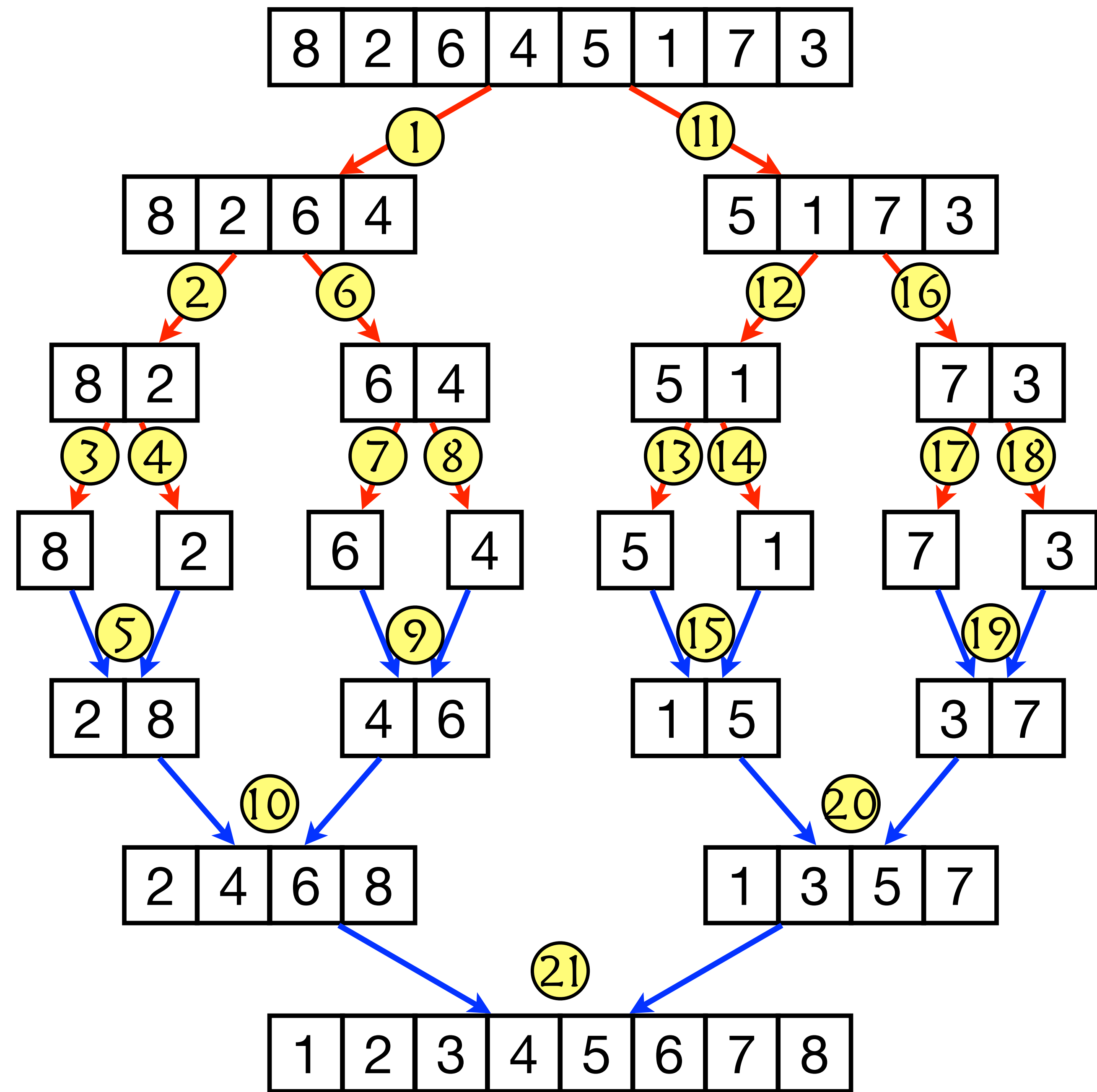
OUTPUT: Sortierter Subarray

MERGE-SORT(A,p,r)

```
1  if  $p < r$ 
2    then  $q \leftarrow \lfloor (p + r)/2 \rfloor$ 
3         MERGE-SORT( $A, p, q$ )
4         MERGE-SORT( $A, q + 1, r$ )
5         MERGE( $A, p, q, r$ )
```



```
1  if  $p < r$ 
2      then  $q \leftarrow \lfloor (p + r) / 2 \rfloor$ 
3           MERGE-SORT( $A, p, q$ )
4           MERGE-SORT( $A, q + 1, r$ )
5           MERGE( $A, p, q, r$ )
```



Subroutine 5.2

INPUT: Zwei sortierte Subarrays von $A=[1,...,n]$, d.h. $A[p,...,q]$ und $A[q+1,...,r]$

OUTPUT: Sortierter Subarray $A[p,...,r]$

MERGE(A, p, q, r)

```

1   $n_1 \leftarrow q - p + 1$ 
2   $n_2 \leftarrow r - q$ 
3  erzeuge die Felder  $L[1..n_1 + 1]$  und  $R[1..n_2 + 1]$ 
4  for  $i \leftarrow 1$  to  $n_1$ 
5      do  $L[i] \leftarrow A[p + i - 1]$ 
6  for  $j \leftarrow 1$  to  $n_2$ 
7      do  $R[j] \leftarrow A[q + j]$ 
8   $L[n_1 + 1] \leftarrow \infty$ 
9   $R[n_2 + 1] \leftarrow \infty$ 
10  $i \leftarrow 1$ 
11  $j \leftarrow 1$ 
12 for  $k \leftarrow p$  to  $r$ 
13     do if  $L[i] \leq R[j]$ 
14         then  $A[k] \leftarrow L[i]$ 
15              $i \leftarrow i + 1$ 
16         else  $A[k] \leftarrow R[j]$ 
17              $j \leftarrow j + 1$ 

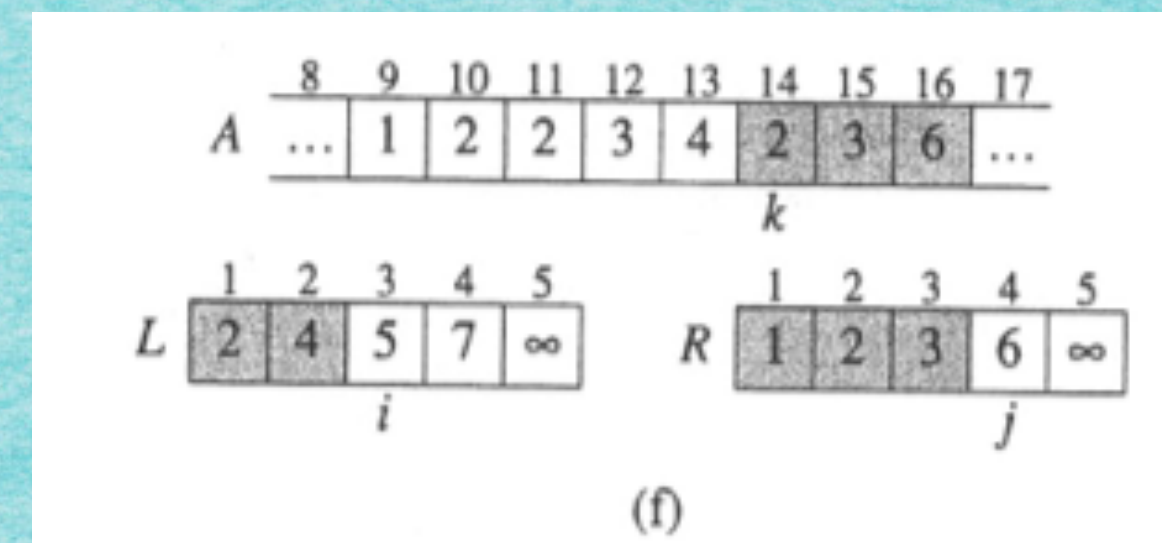
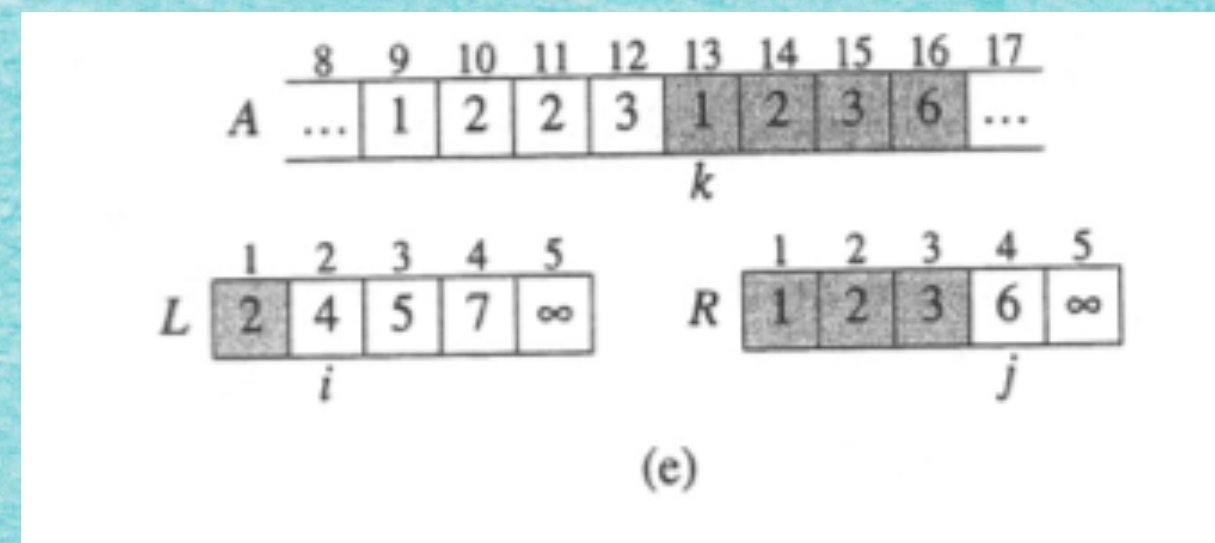
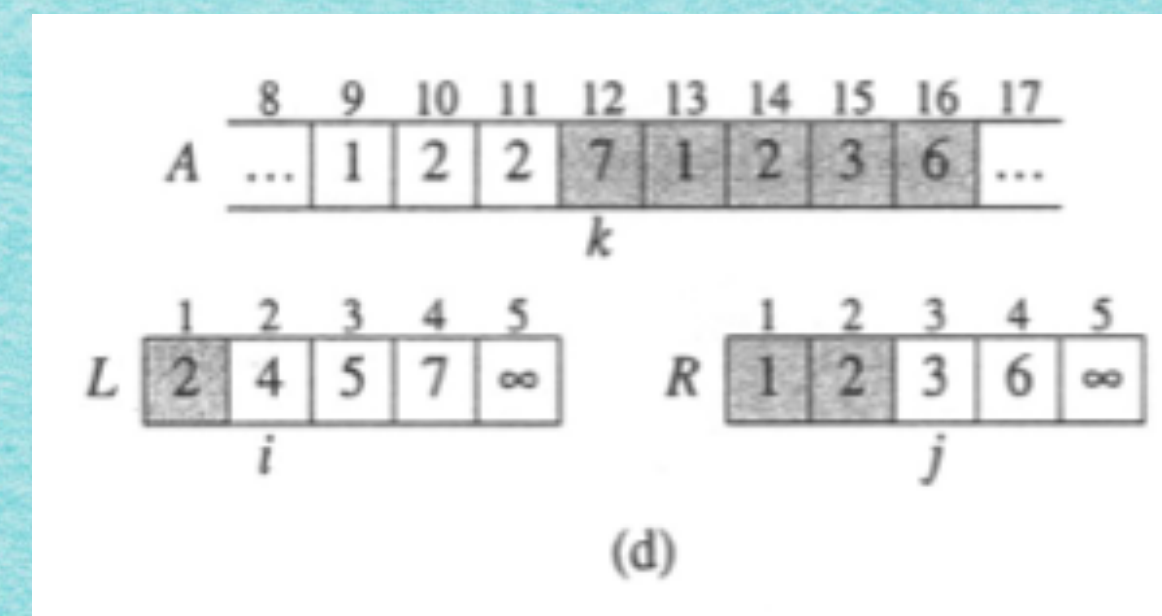
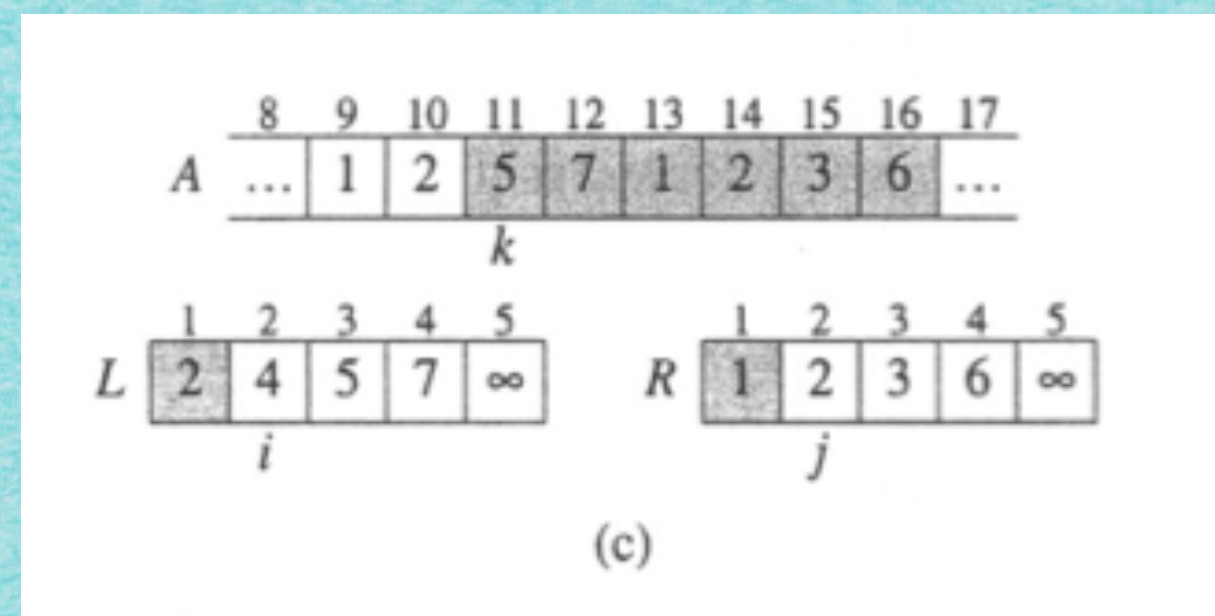
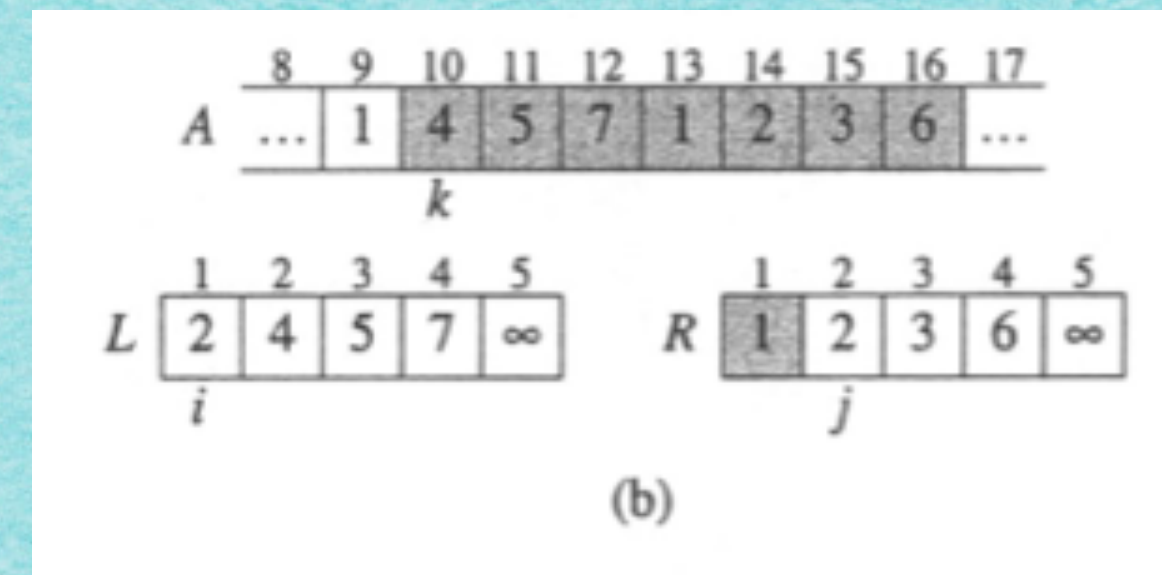
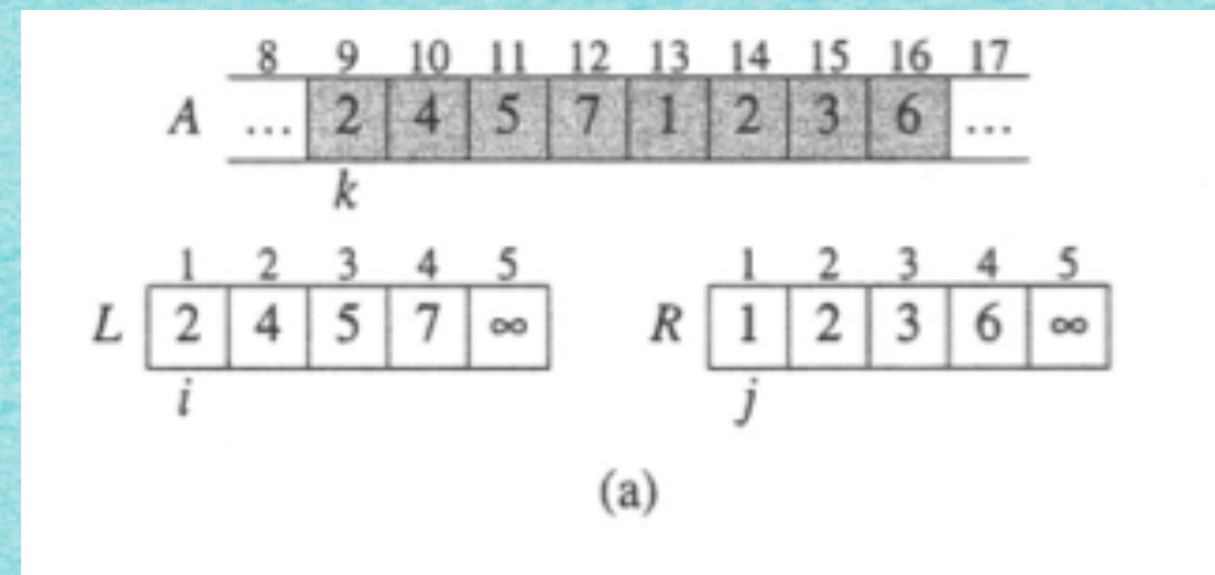
```

	8	9	10	11	12	13	14	15	16	17
A	...	2	4	5	7	1	2	3	6	...

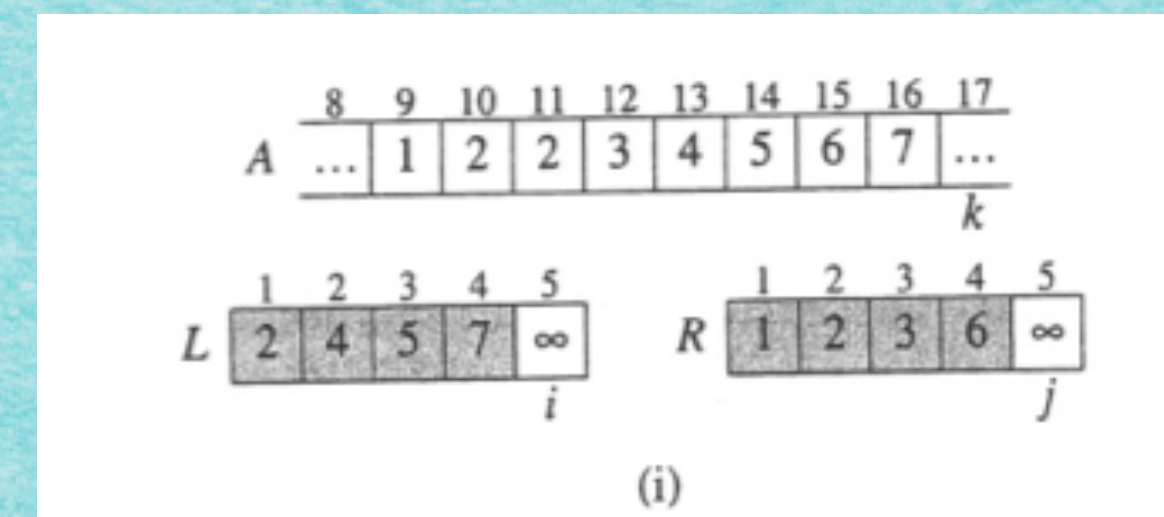
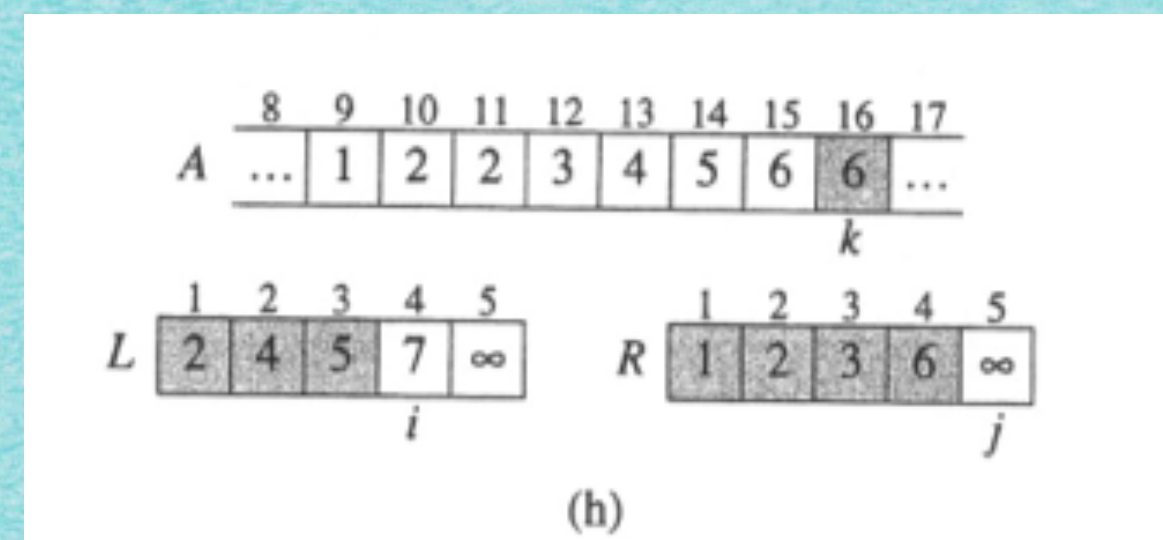
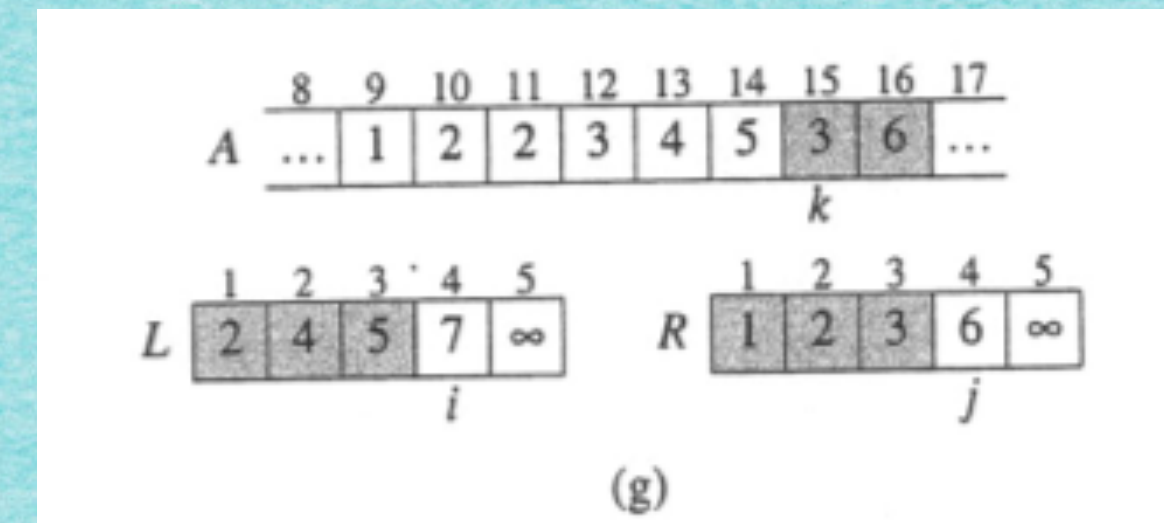
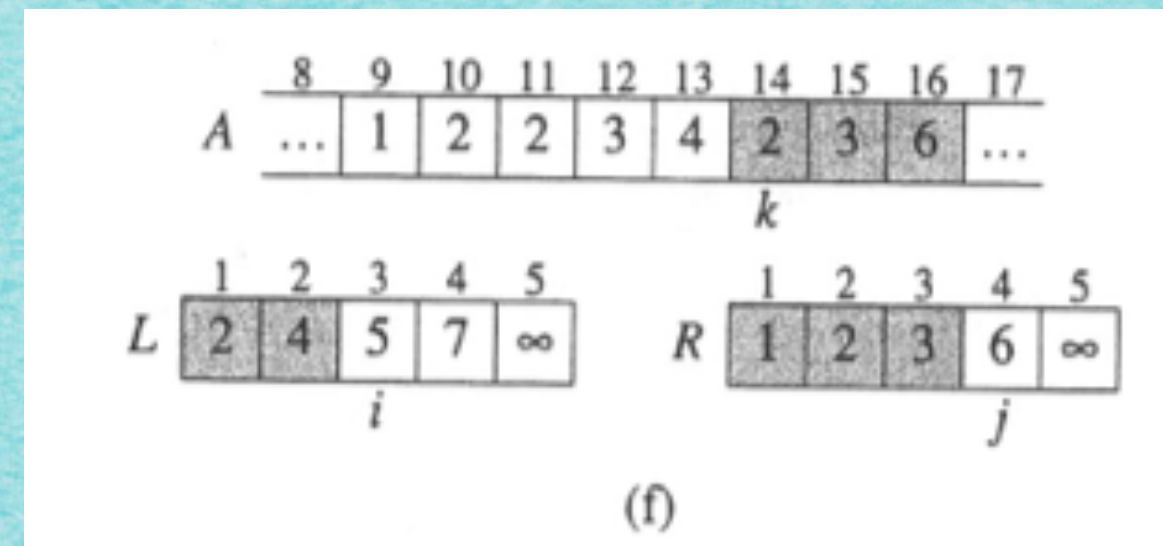
	8	9	10	11	12	13	14	15	16	17
A	...	2	4	5	7	1	2	3	6	...
		2	3	4	5					
		2	3	4	5					
		2	3	4	5					

	8	9	10	11	12	13	14	15	16	17
A	...	1	2	5	7	1	2	3	6	...
		2	3	4	5					
		2	3	4	5					
		2	3	4	5					

5.1 Mergesort



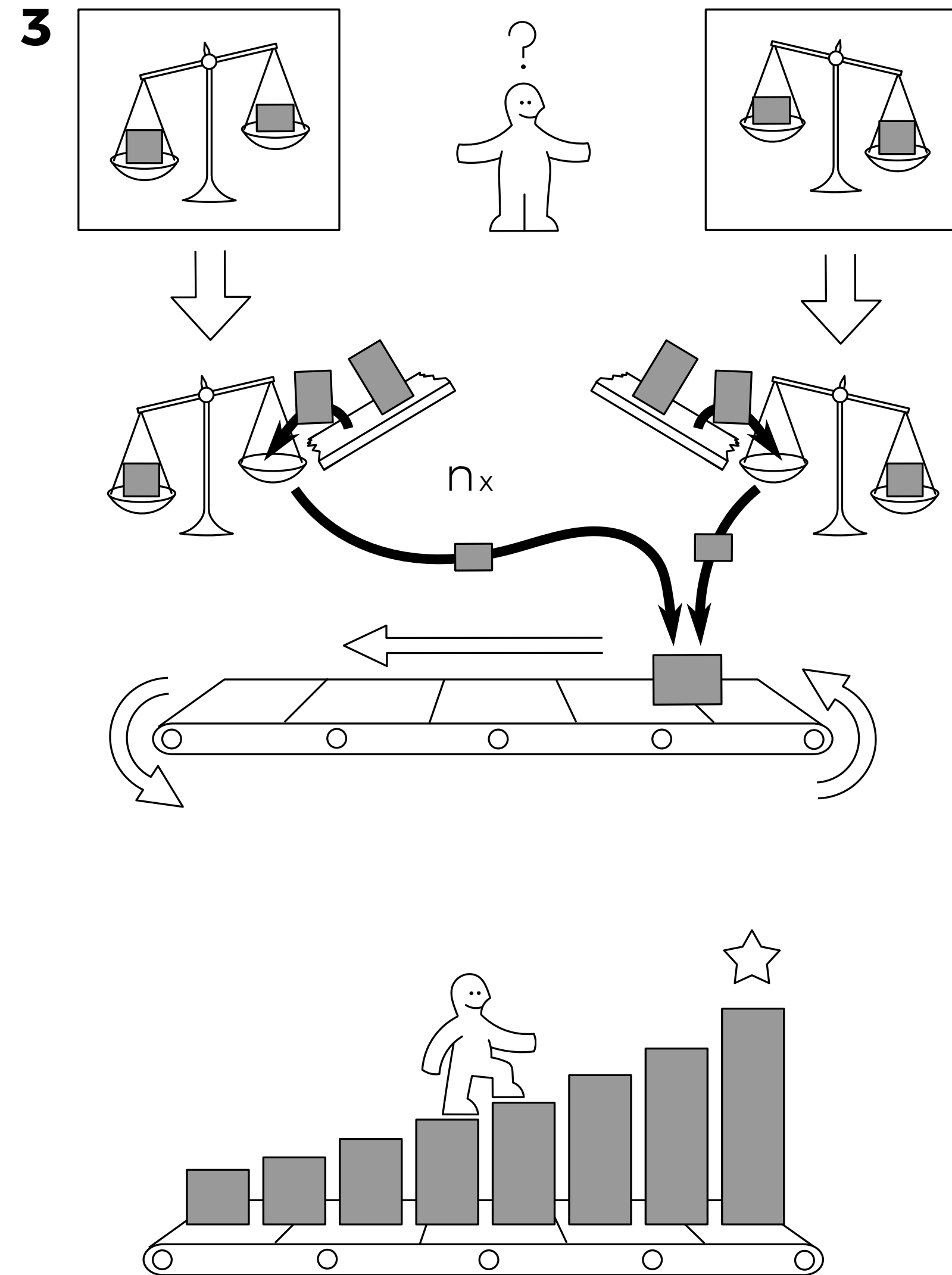
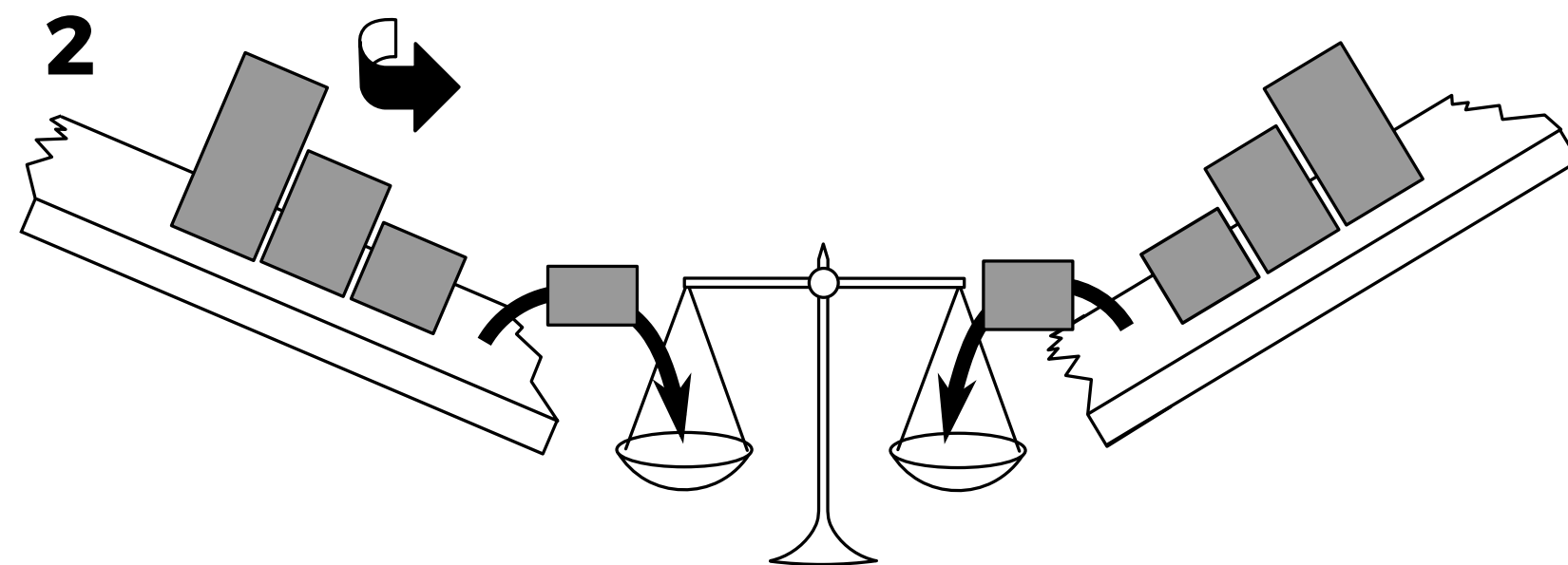
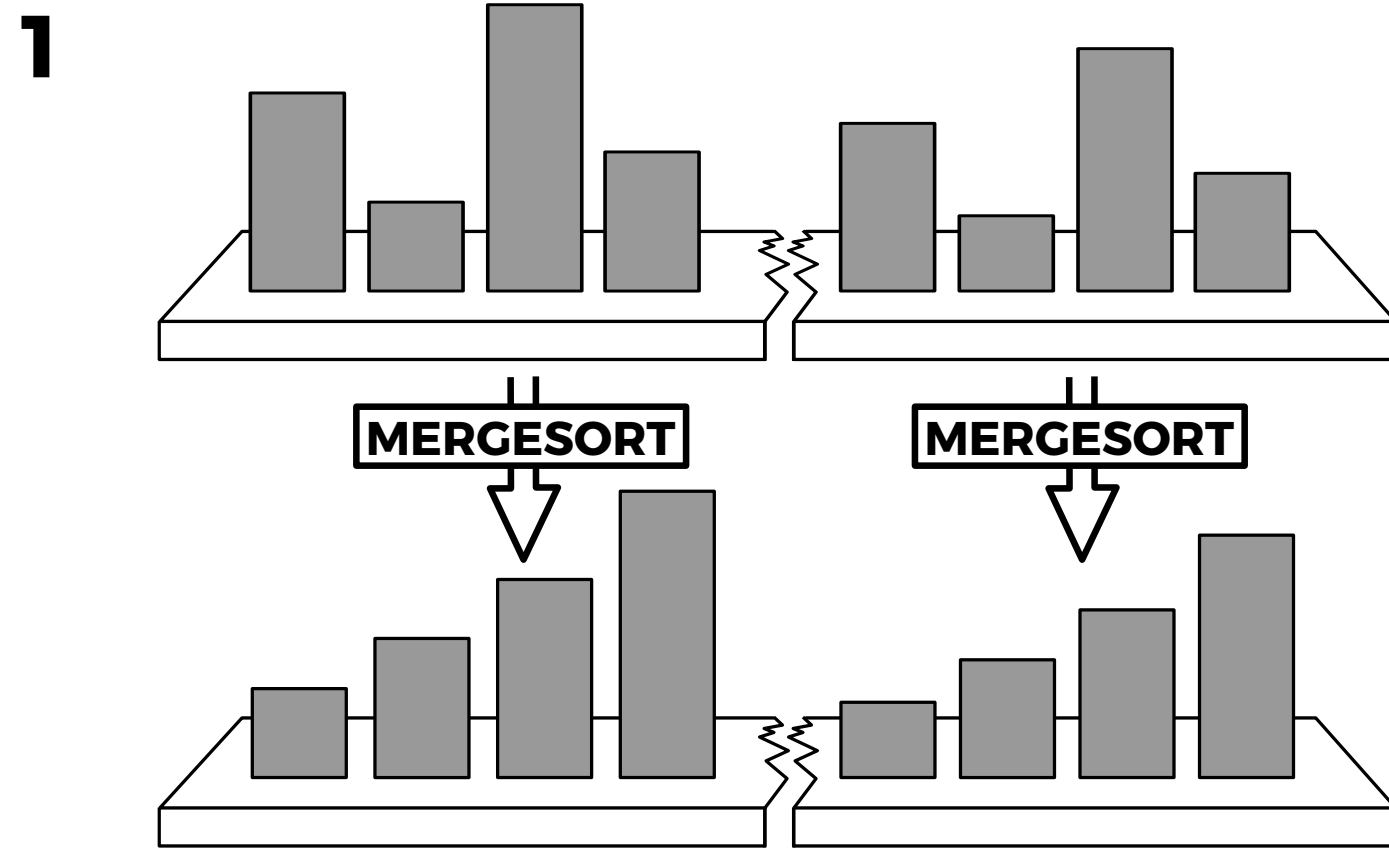
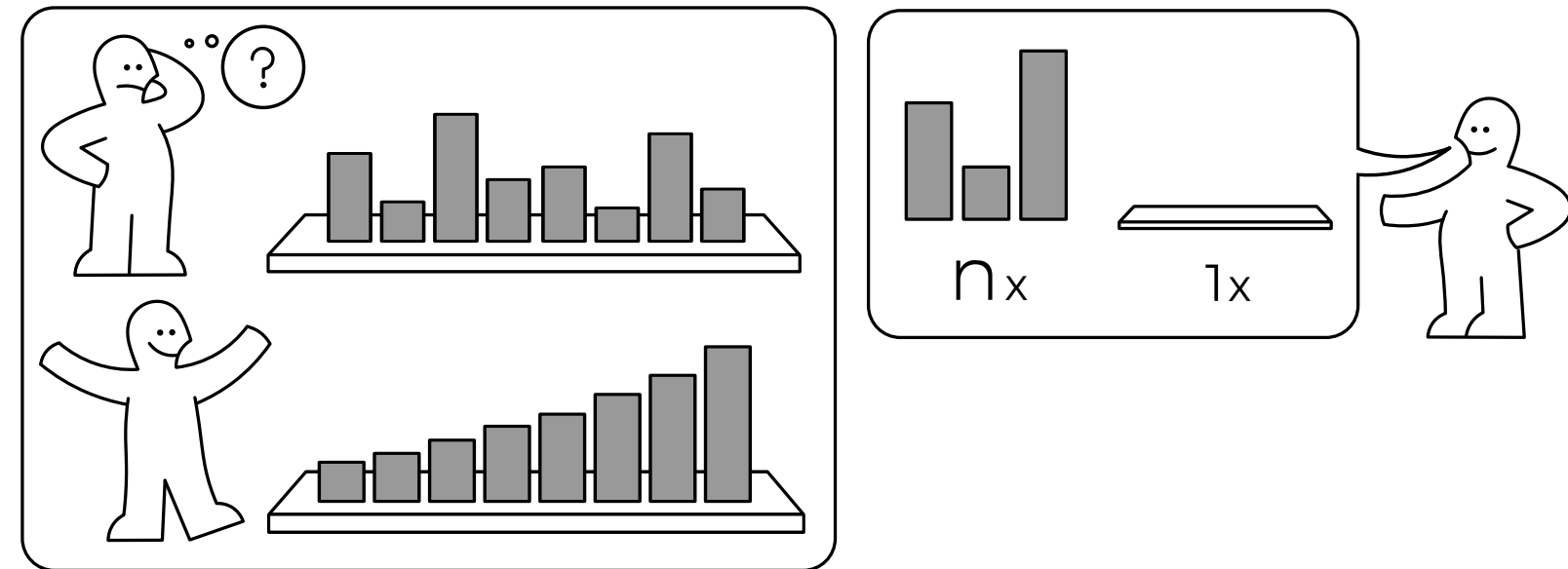
5.1 Mergesort



MERGE SÖRT

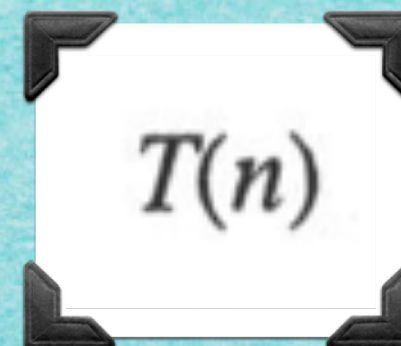
idea-instructions.com/merge-sort/
v1.0, CC by-nc-sa 4.0

IDEA



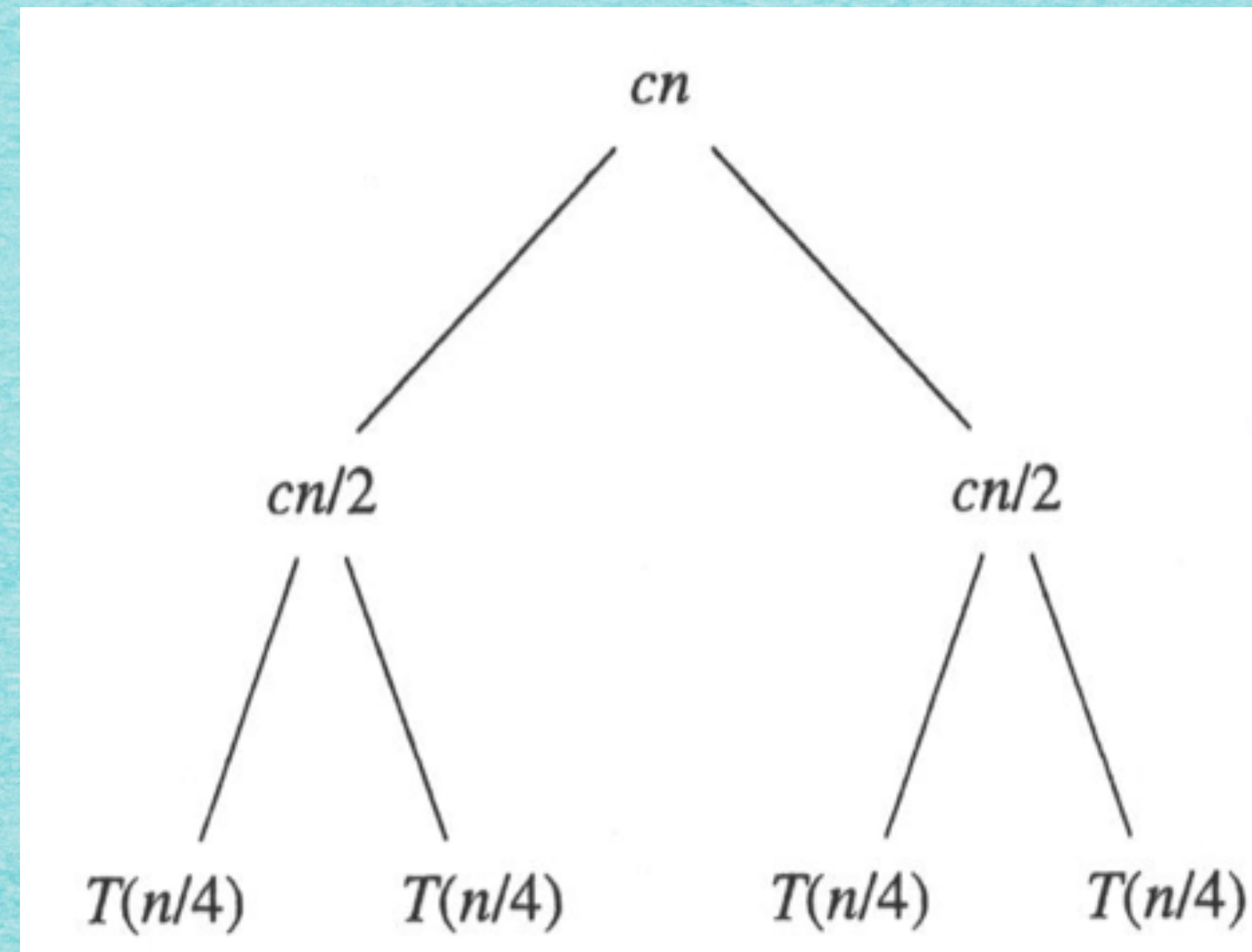
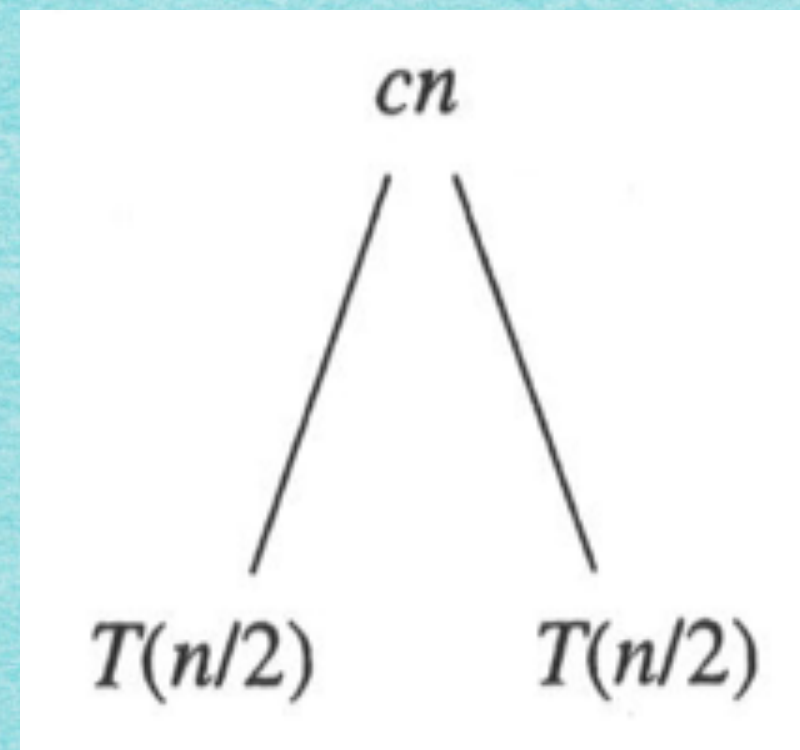
5.1.3 Laufzeit von Mergesort

Wie viele Schritte benötigt Merge-Sort für einen Array der Länge n ?

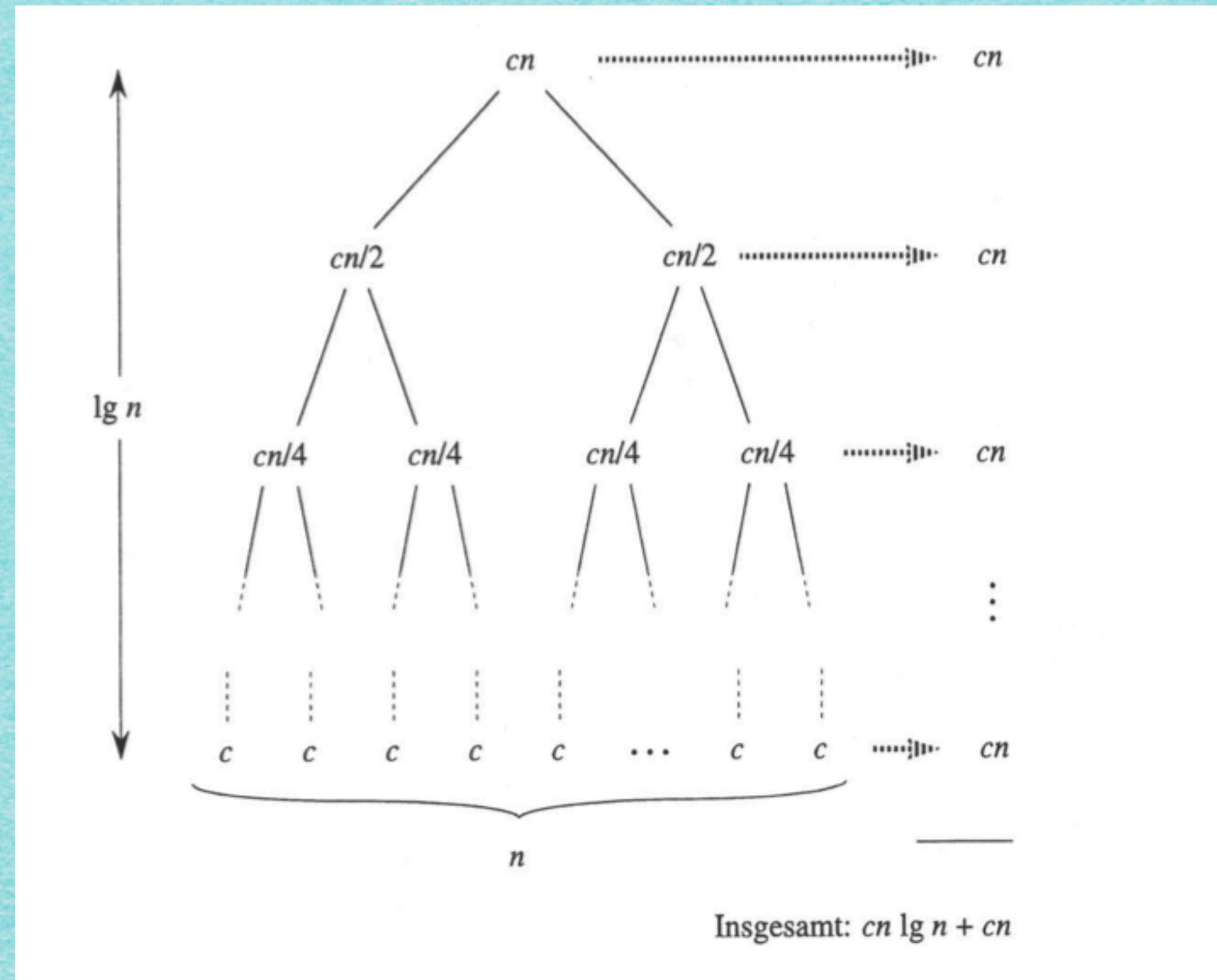
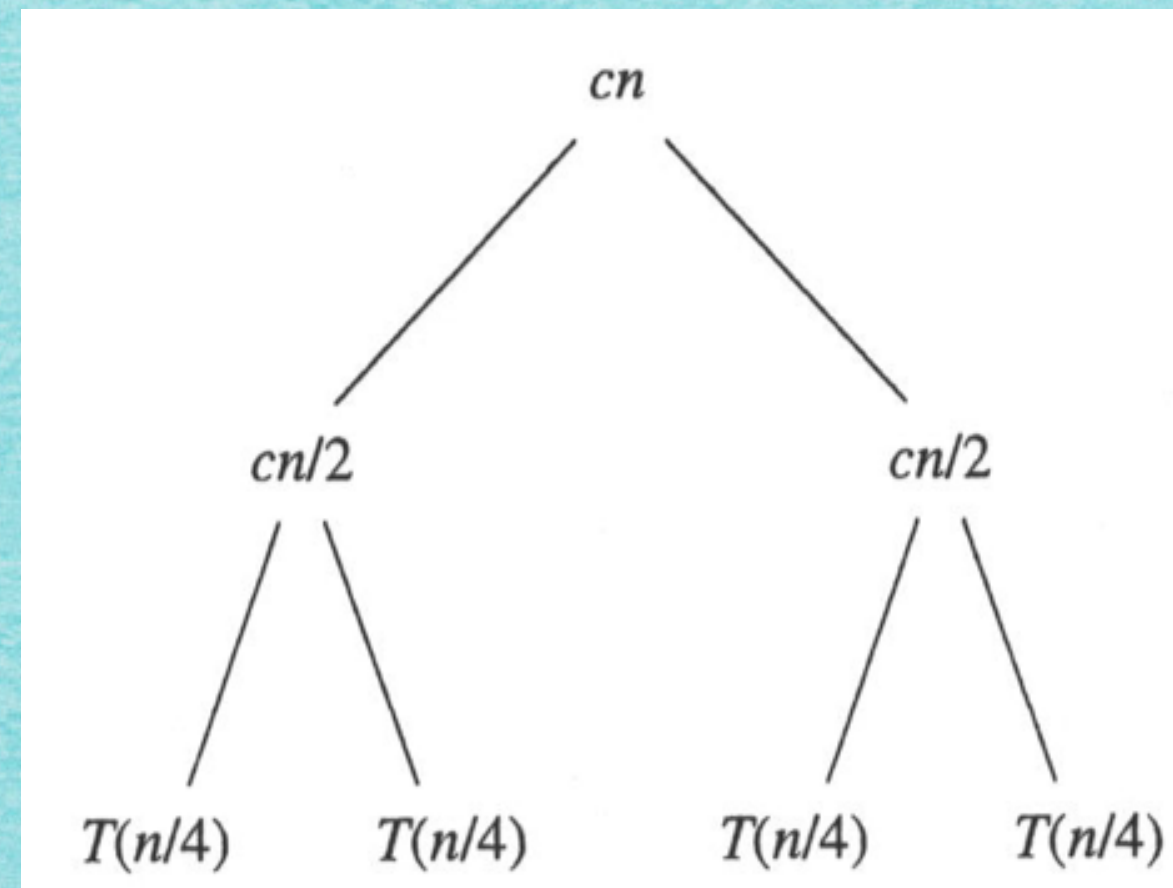

$$T(n)$$

5.1.3 Laufzeit von Mergesort

$T(n)$



5.1.3 Laufzeit von Mergesort



5.1.3 Laufzeit von Mergesort

Satz 5.3 (Komplexität von Mergesort)
Für einen n -elementigen Array A hat Mergesort eine Laufzeit von $O(n \log n)$.

Mehr an der Tafel!

s.fekete@tu-bs.de