

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

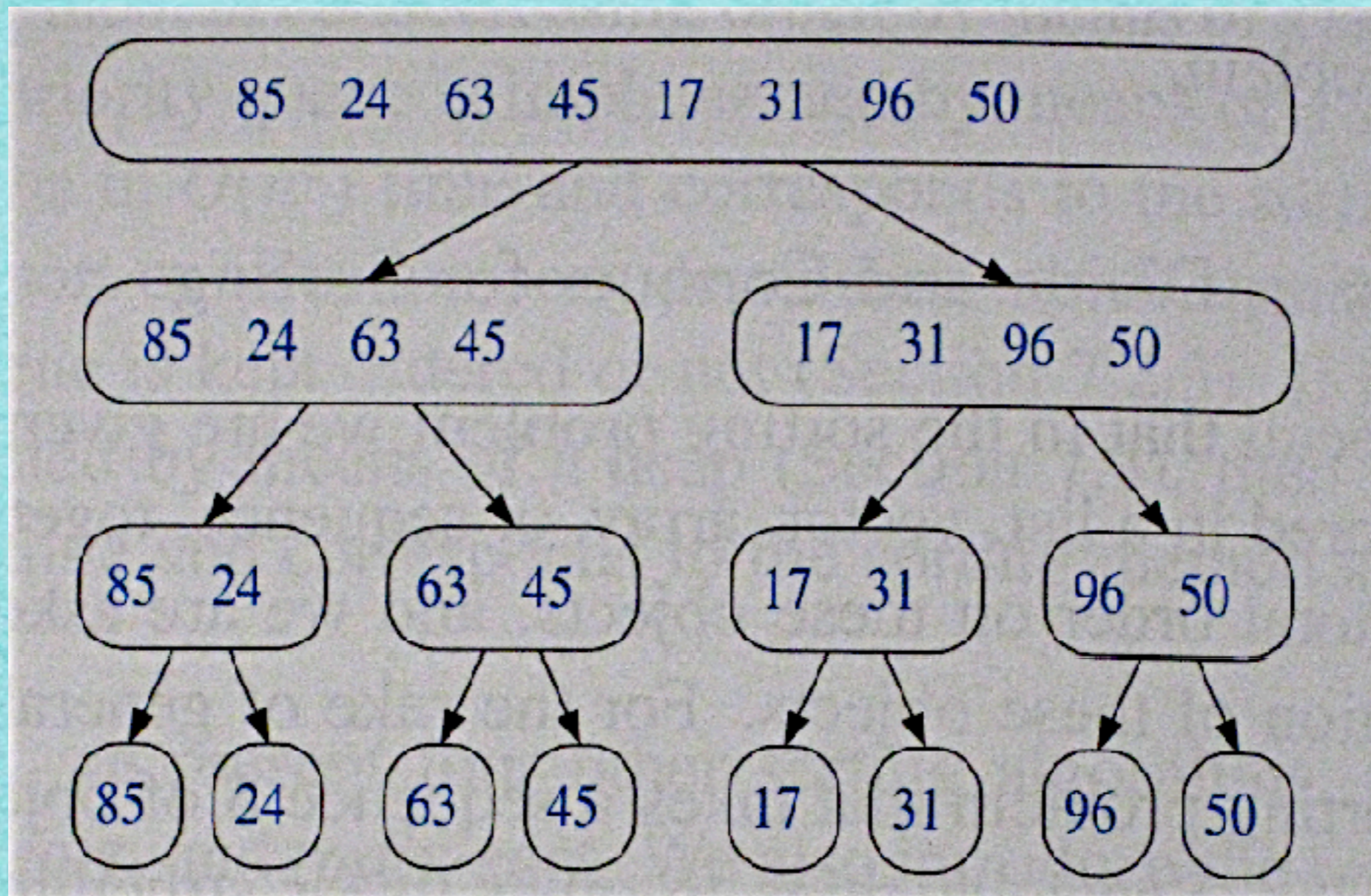


Kapitel 5: Sortieren

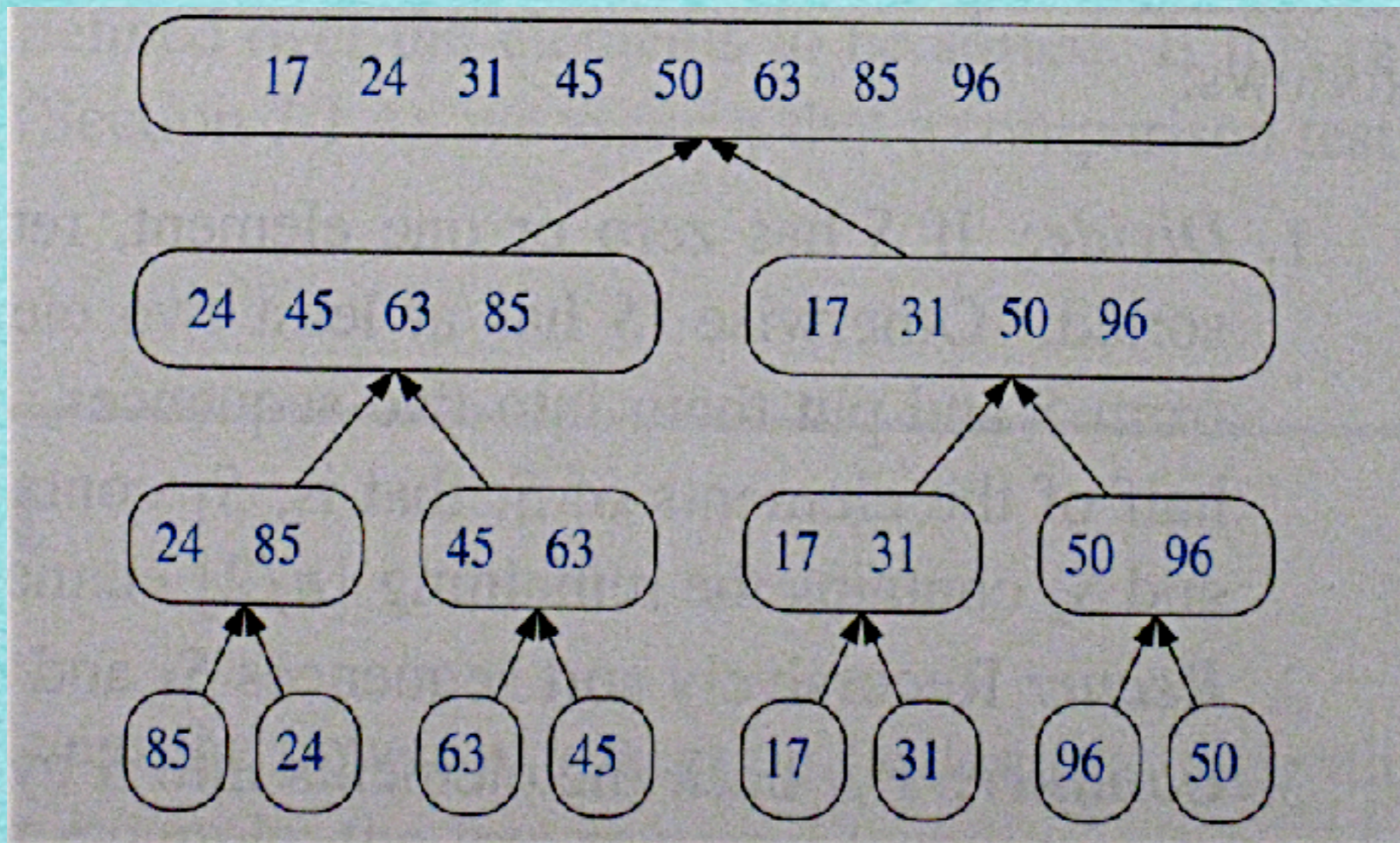
*Algorithmen und Datenstrukturen
WS 2020/21*

Prof. Dr. Sándor Fekete

5.2 Mergesort



5.2 Mergesort



5.2.2 Algorithmische Beschreibung

Algorithmus 5.1

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

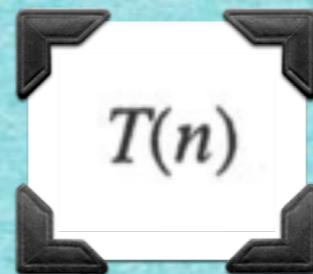
OUTPUT: Sortierter Subarray

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( $A, p, q, r$ )
```

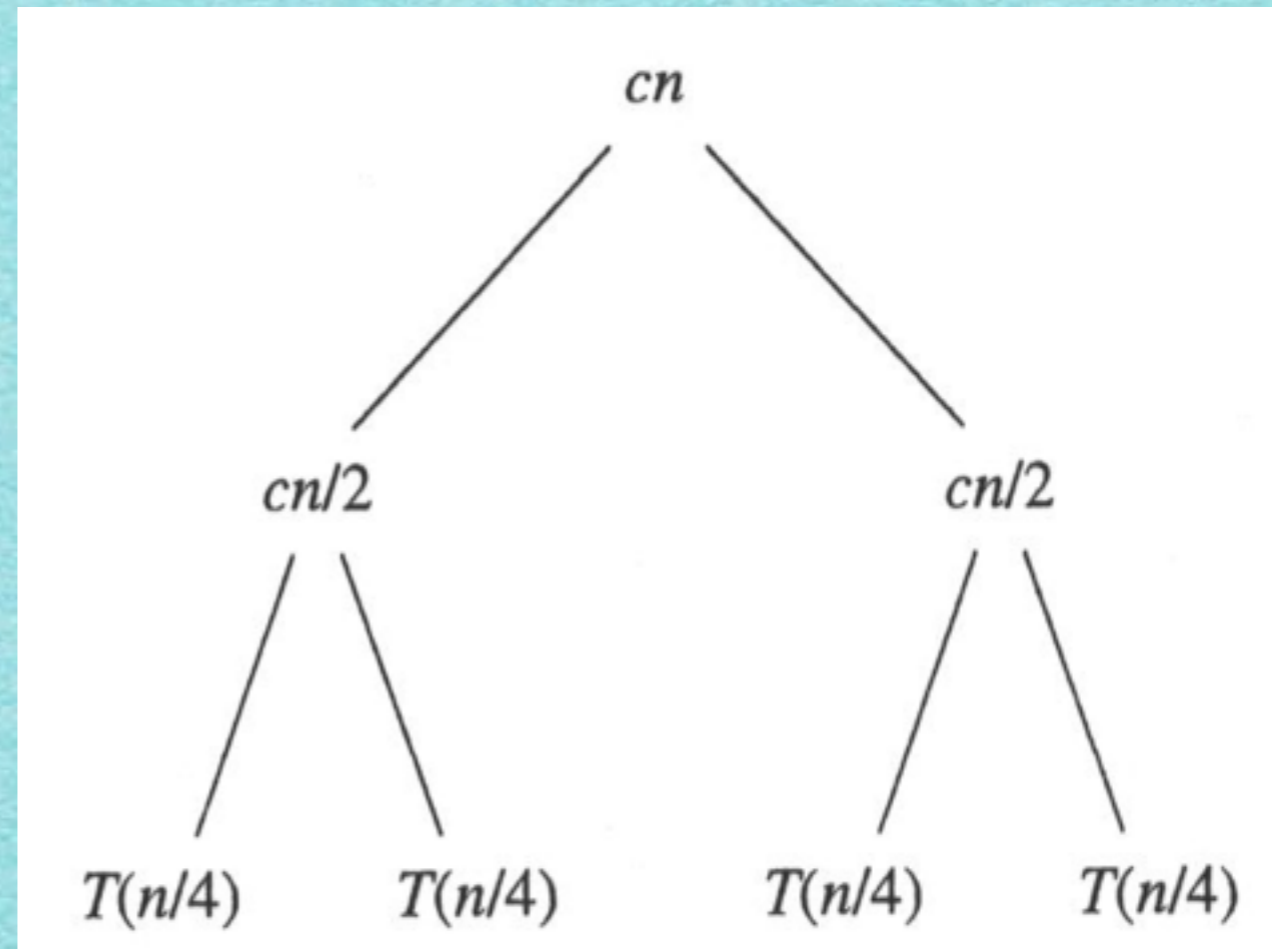
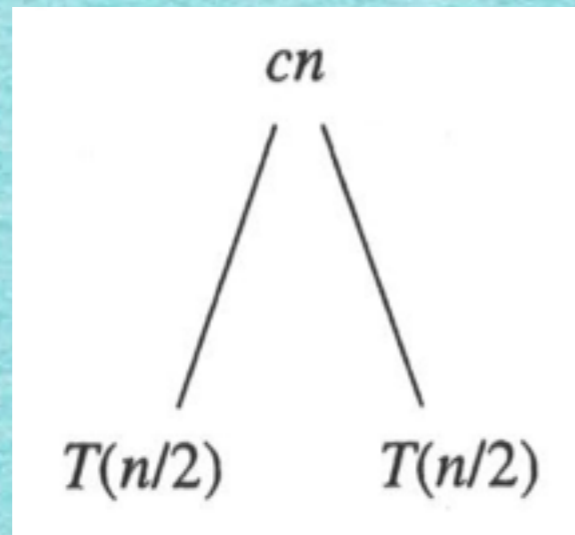
5.2.3 Laufzeit von Mergesort

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

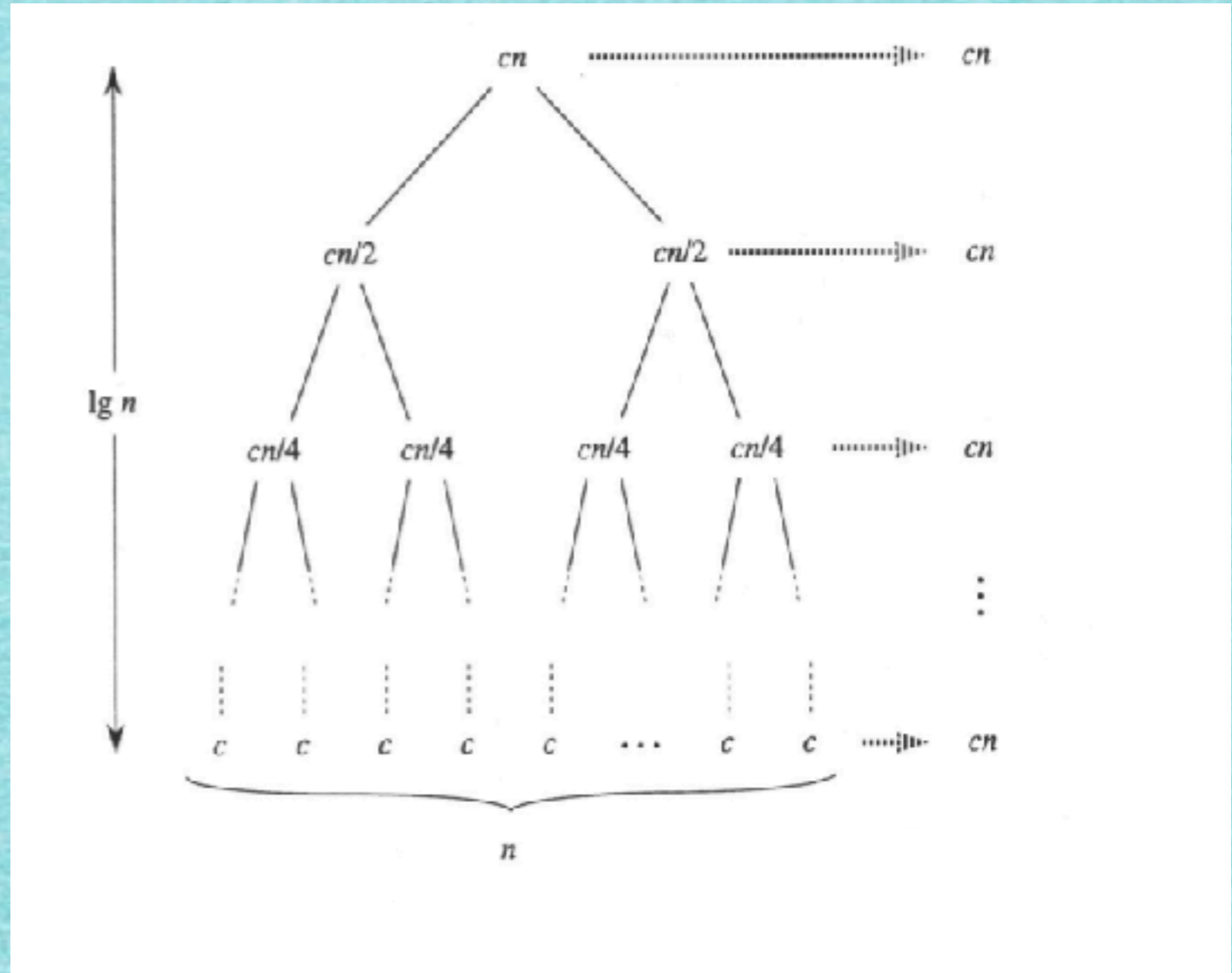
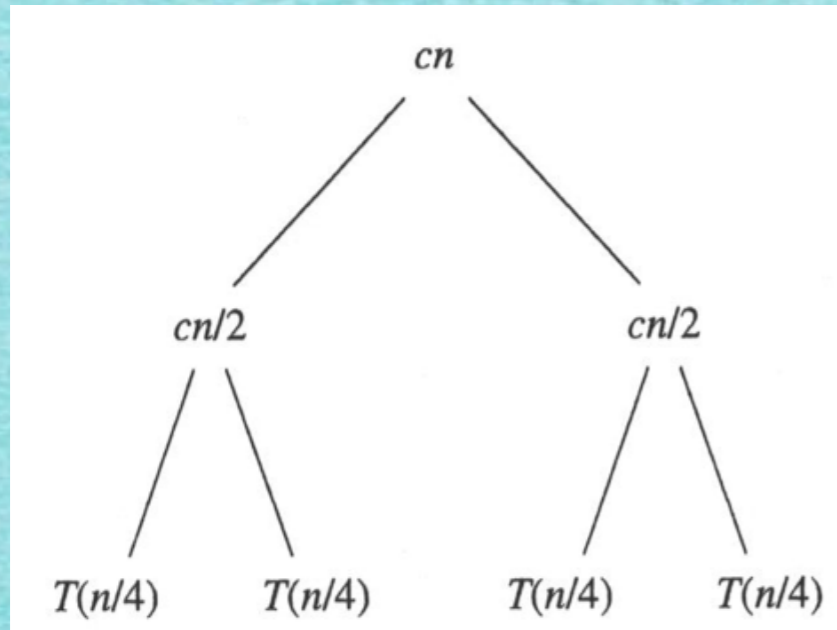


5.2.3 Laufzeit von Mergesort

$T(n)$



5.2.3 Laufzeit von Mergesort



5.2.3 Laufzeit von Mergesort

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

5.2.3 Laufzeit von Mergesort

Fragen:

- Geht's noch schneller?
- Wie kann man sonst mit Rekursionen umgehen?

Mehr an der Tafel!

s.fekete@tu-bs.de