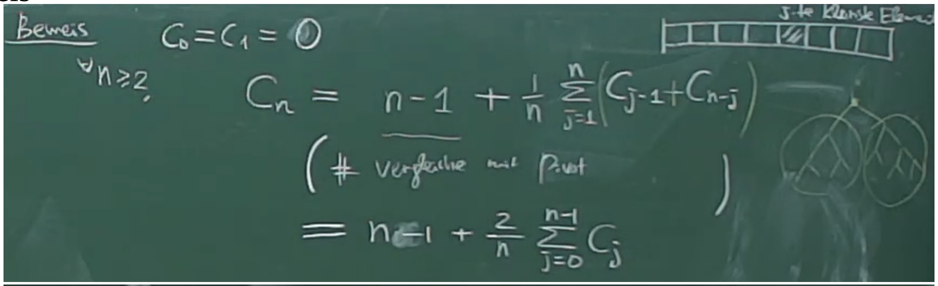
Übersicht

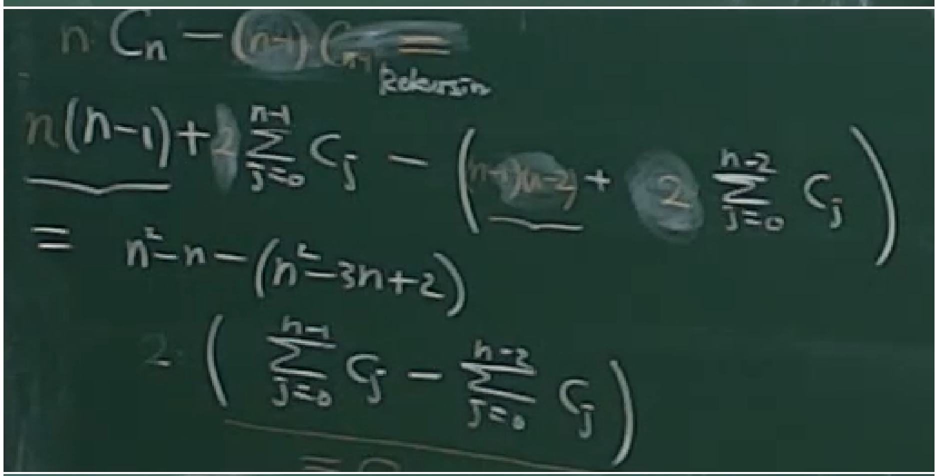
- [[Sortierverfahren]]
- gegeben Array von unterschiedlichen Elementen
- Divide and Conquer
  - in zwei Subarrays aufteilen
  - Teilarrays rekursiv sortieren
  - sortierte Arrays zusammenfügen
- Input
  - Array a
  - linke/kleinste Index l
  - rechte/größte Index r
- zwischen l und r wird sortiert

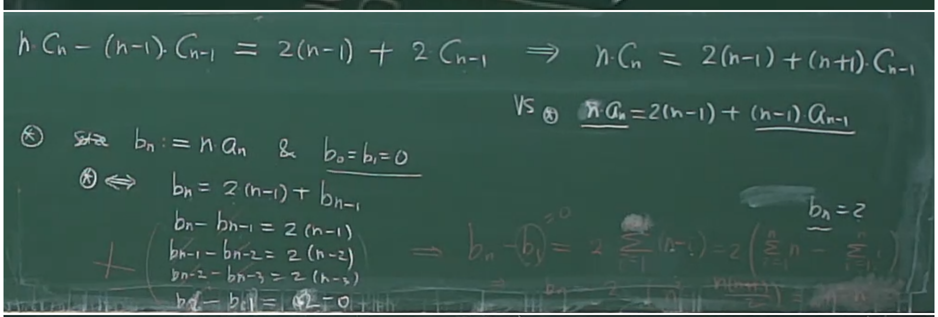
Algorithmus

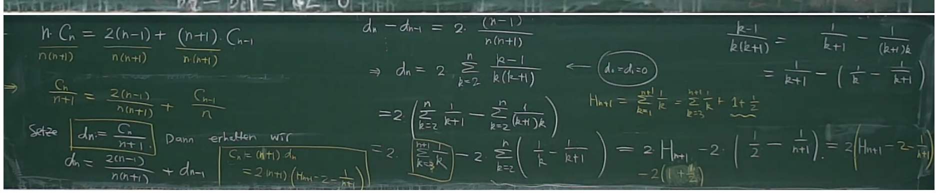
- l=1 und r=n beim ersten Aufruf
- p:=a[r] ist Pivot-Element
- vergleiche a[l] bis a[r-1] mit p
- sortiere um, sodass
  - a[k]=p
  - a[i]<p für i<k
  - a[i]>p für i>k
- ruf Quicksort rekursiv für linke und rechte Seite auf
  - links mit den Parametern
    - l=l, r=k-1
  - rechts mit den Parametern
    - l=k+1, r=r

- angenommen alle Eingabepermutationen sind gleich wahrscheinlich
- C_n ist Anzahl der Vergleiche für Array mit n Elementen
  - $C_n = 2(n+1)(H_{n+1} - 2 + \frac{1}{n+1})$
  - $H_n := \sum_{k=1}^{n} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{n}$
    - n-te harmonische Zahl
- $H_n - ln(n) - \gamma$ konvergiert gegen 0 für n->∞
  - γ:=Euler-Mascheroni-Konstante
  - ~0.577
- $\frac{C_n}{2nln(n)}$ konvergiert gegen 1 asymptotisch

- Beweis
  - 

  - 

  - 

  - 

[[Nicht Lineare Rekursion]]