

Untere Schranke

- jedes vergleichsbasierte [[Sortierv Verfahren]] braucht im worst-case zumindest
 - $c * n * \log(n)$
 - $\Omega(n * \log(n))$

Die Höhe eines Binärbaums mit $n!$ Blättern ist $\Omega(n \log n)$

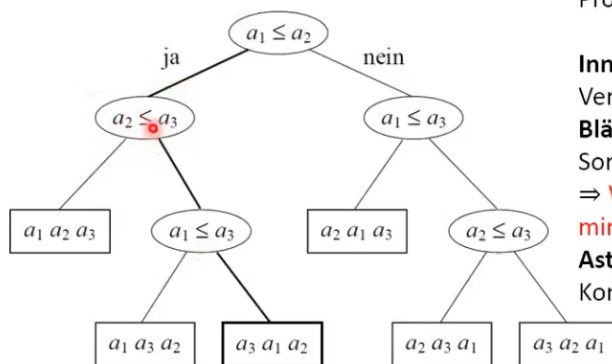
$\Rightarrow \Omega(n \log n)$ ist eine untere Schranke für die Anzahl der im **worst case** zum Sortieren notwendigen **Vergleiche**

\Rightarrow Die **worst case Laufzeit vergleichsorientierter Sortierv Verfahren** ist $\Omega(n \log n)$

- MergeSort ist **worst-case optimal**

Entscheidungsbaum von vergleichsbasierten Sortierv Verfahren

Beispiel: InsertionSort auf a_1, a_2, a_3 :



alle möglichen
Programmverzweigungen

Innere Knoten:

Vergleiche zwischen Elementen

Blätter:

Sortierte Reihenfolge des Inputs

\Rightarrow **Wie viele Blätter gibt es mindestens?**

Ast:

Kontrollfluss für best. Input

- \swarrow Ast für $a_1=5, a_2=8, a_3=4$
- worst-case Verhalten = längste Ast
 - $|Knoten| = |Vergleiche|$
- längste Ast kürzestmöglich, wenn alle Äste möglich lang
- idealer Algorithmus
 - vollständiger Binärbaum mit $n!$ Blätter