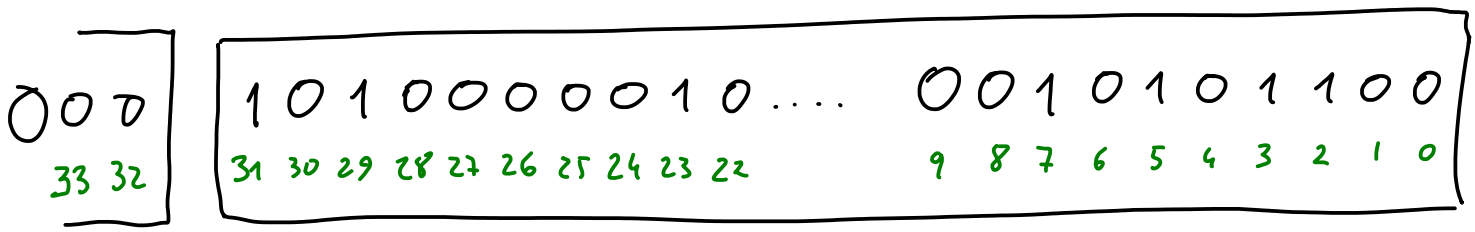
Untere Schranken

vergleichsbasierte Datenstrukturen/Algorithmen

Erlaubt: Vergleich zwischen Schlüssel

Gegenbeispiel: Bitvektor für Mengen



$\{2, 3, 5, 7, \ldots 23, 29, 31\}$

Einfügen, Löschen, enthulten? ... $O(1)$

! eingeschränkte Grundmenge

einfügen(24): `M = M | 1<<24;`

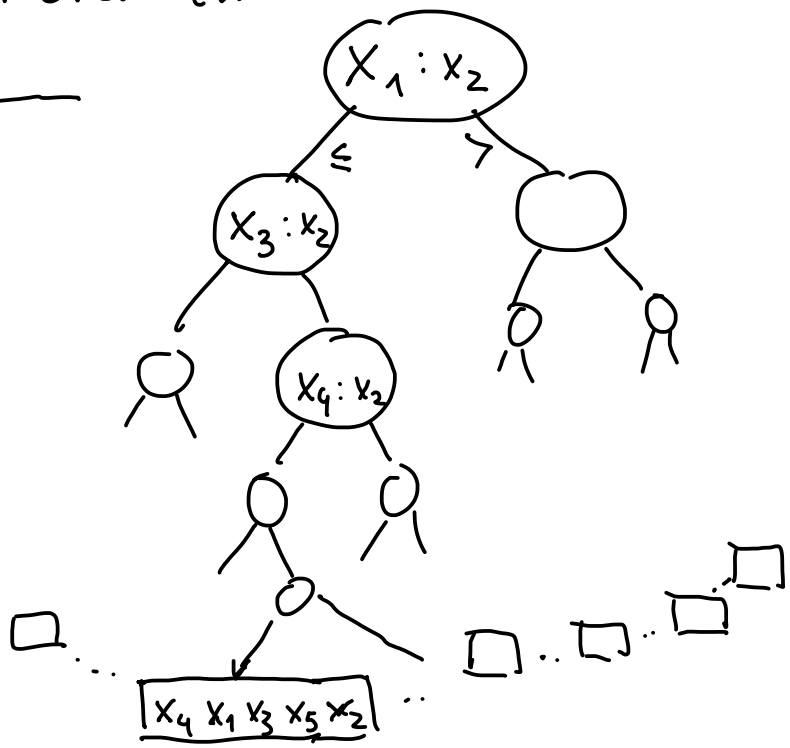
---

anderes Gegenbeispiel: Hashtabelle

untere Schranke für Sortieren im vergleichsbasierten Modell

abstraktes Modell:

Entscheidungsbaum



x_1, x_2, x_3, x_4, x_5

$\geq n!$ Blätter, Binärbaum

Höhe $\geq \log_2 n!$

$\geq \log\ (1 \cdot 2 \cdot 3 \cdot \ldots \cdot \frac{n}{2} \cdot \frac{n}{2}+1 \cdot \ldots \cdot n)$

(≥ 1 für die ersten Faktoren, $\geq n/2$ für die letzten Faktoren)

$$\geq \log\left(\left(\frac{n}{2}\right)^{n/2}\right) = \frac{n}{2} \cdot \log_2 \frac{n}{2} = \frac{n}{2}(\log_2 n - 1) = \Omega(n \log n)$$

SATZ: Sortieren von n Elementen mit einem vergleichsbasierten Algorithmus benötigt mindestens $\log_2 n! = \Omega(n \log n)$ Vergleiche.

Vergleichsbasierte Datenstrukturen für das Wörterbuchproblem können typischerweise auch

- das kleinste Element bestimmen
- statt eines nicht vorhandenen Schlüssels x den nächstkleineren / nächstgrößeren finden.

Wenn man in $O(f(n))$ Zeit einfügen, löschen, suchen, und das kleinste Element finden kann,

dann kann man in $O(n \cdot f(n))$ Zeit sortieren

$\Rightarrow f(n) = \Omega(\log n)$.