

Radixsort stellt **besondere Anforderungen an die zu sortierenden Elemente** (es muss eine **keine beliebige beliebige** Ordnung für die Elemente existieren) - die zu Beginn dieses Kapitels eingeführte Verallgemeinerung mit Hilfe der `compareTo`-Methode, beliebige Objekte "sortierbar zu machen", funktioniert für Radix-Sort nicht! Dafür hat Radixsort eine Zeitkomplexität von ca. $O(n)$ - das lässt sich bei vergleichsbasierten Sortierverfahren nicht erreichen. Radixsort ist ein **nicht-vergleichender** Sortieralgorithmus. Er funktioniert besonders gut beim Sortieren von Ganzzahlen, darum betrachten wir hier zunächst auch nur diesen Fall.

Weitere Wichtige Eigenschaften:

Radixsort sortiert die Elemente einer Liste, indem er sie nach einzelnen Ziffern (Stellenwerten) gruppiert, meistens beginnend mit der niederwertigsten Ziffer. Für jede Ziffer der zu sortierenden Ziffernpositionen.

- **Nicht-vergleichend:** Radixsort vergleicht keine Elemente direkt miteinander, sondern nutzt die Ziffernpositionen.
- Bei entsprechender Implementierung ist Radixsort stabil: Die Reihenfolge gleichwertiger Elemente bleibt wie in der Ausgangsliste erhalten.
- **Laufzeit:** Die Zeitkomplexität beträgt im Allgemeinen $O(k \cdot (b+1) \cdot n)$, wobei k die maximale Schlüssellänge (Anzahl der Ziffern), b die Basis (z.B. 10 bei Dezimalzahlen) und n die Anzahl der zu sortierenden Elemente ist.
- Radixsort ist besonders effizient, wenn die Anzahl der Ziffern (bzw. die Schlüssellänge) im Vergleich zur Anzahl der Elemente klein ist.

From:
<https://www.info-bw.de/> -

Permanent link:
<https://www.info-bw.de/faecher:informatik:oberstufe:algorithmen:sorting:radixsort:start?rev=1751359699>

Last update: **01.07.2025 08:48**

