

Programmieren in C

Sorting

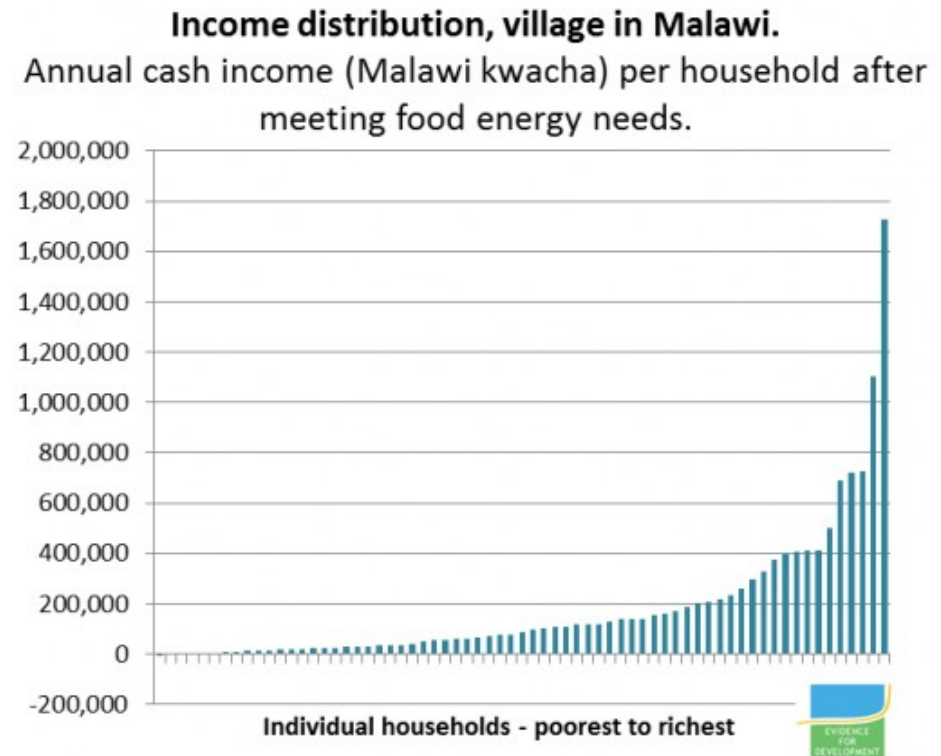
Dr. Adrian Koller, Büro E307, adrian.koller@hslu.ch

Warum sortieren?

- Grosse Datenmengen wollen analysiert werden

→ Aus Abfolgen/
Reihenfolgen lassen
sich Muster erkennen

→ Filtern nach Kriterien



So schnell wie möglich...

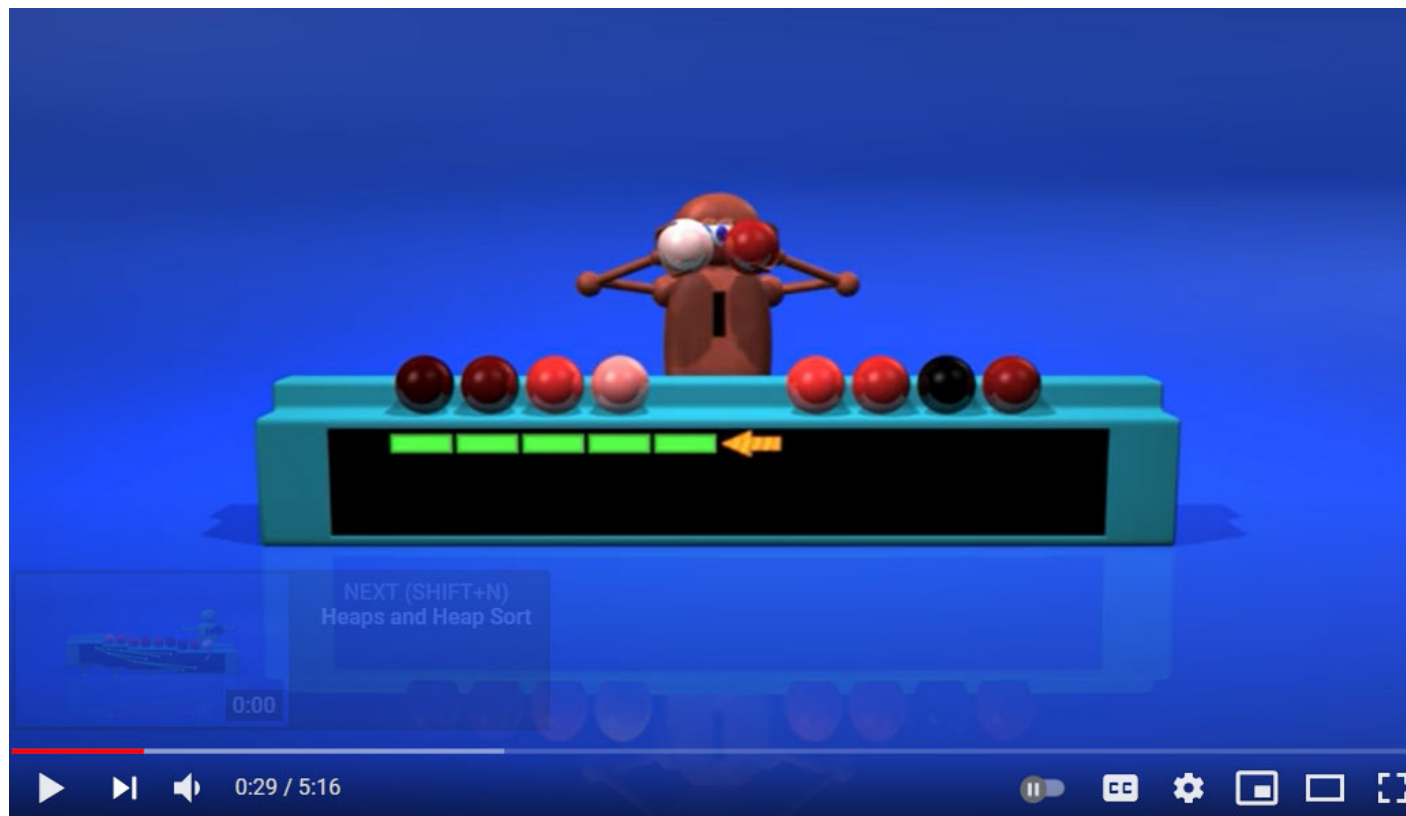
- Daten sortieren braucht Zeit → Zeit = Geld...
→ So schnell wie möglich sortieren!

QuickSort, MergeSort, TimSort, HeapSort, Bubble Sort, Insertion Sort, Selection Sort, Tree Sort, Shell Sort, Bucket Sort, Radix Sort, Counting Sort, CubeSort...

Bewertung: - Zeitbedarf (time complexity)
 - Platzbedarf (space complexity) → Speicherbedarf

Beispiel: Insertion Sort

- <https://www.youtube.com/watch?v=TZRWRjq2CAg>



Insertion Sort → Repo

- 40_Exercises/13_sorting
→ insertionsort.c

1. Verstehen
2. Anpassen, so dass aufsteigend oder absteigend sortiert werden kann
3. Funktioniert «Insertion Sort» auch mit floats? Warum ja, warum nein?

Weitere Sorting Algos

- <https://www.edureka.co/blog/sorting-algorithms-in-c/>
- Code kopieren und zum Laufen bringen
 - kommentieren/verstehen
 - eigene Anpassungen
- «grosses» Array mit zufälligen Zahlen erzeugen
 - sortieren, Zeit messen, vergleichen...