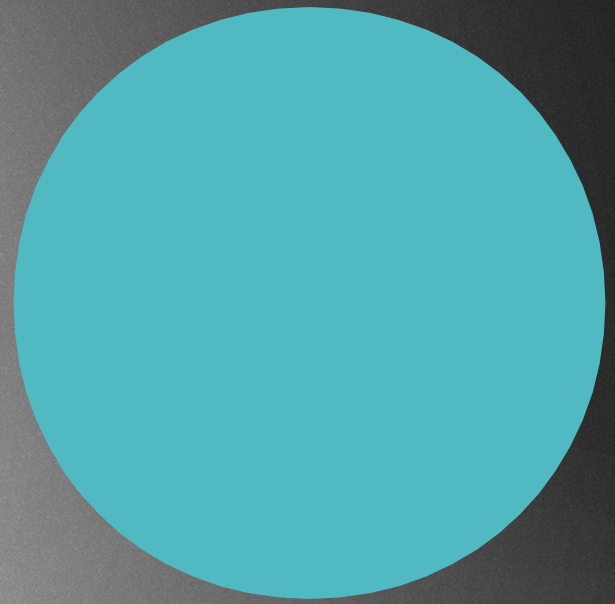
HYBRID ALGORITHMS



Introsort and timsort

Hybrid algorithms

- It combines more algorithms to solve a given problem
- It choses one algorithm depending on the data or switching between them over the course of the algorithm
- This is generally done to combine desired features of each, so that the overall algorithm is better than the individual components
- Important: hybrid algorithm does not refer to simply combining multiple algorithms to solve a different problem but only to combining algorithms that solve the same problem → but differ in other characteristics // such as performance
- The technique can be used when sorting

Hybrid algorithms

- Heapsort → it has an advantage of a guaranteed running time **O(N logN)**
- Quicksort → optimal implementations are outperform both mergesort and heapsort
- **BUT** quicksort can have quadratic running time when we keep choosing „bad” pivots
- Solution: let’s combine the two algorithms

Introsort

- Also known as introspective sort
- It is a hybrid sorting algorithm that provides both fast avarage performance and optimal worst-case performance
- It begins with quicksort and switches to heapsort when quicksort becomes too slow

INTROSORT = QUICKSORT + HEAPSORT

Hybrid algorithms

- Insertion sort: very efficient on small data (**5 - 10** elements)
- Mergesort / quicksort: asymptotically optimal on large datasets, but the overhead becomes significant if applying them to small datasets
- Solution: let's combine the two algorithms
- Highly optimized hybrid algorithm: timsort

TIMSORT = INSERTION SORT + MERGESORT

Timsort

- Combines mergesort and insertion sort
- It is a stable sorting algorithm
- It was implemented by Tim Peters in 2002 for use in the Python programming language
- Best case running time: **O(N)**
- Worst case running time: **O(N logN)**
- Worst case space complexity: **O(N)**