Freie Universität Berlin

Prof. Dr. Claudia Müller-Birn, Barry Linnert

Objektorientierte Programmierung, SoSe 17

Übung 06

TutorIn: Thierry Meurers
Tutorium 10

Stefaan Hessmann, Jaap Pedersen, Mark Niehues

13. Juni 2017

---

1 Aufgabe 1

a) Die von Python bereitgestellte sorted()-Funktion nutzt einen hybriden Sortier-Algorithmus (Mix aus Mergesort und Insertionsort) namens Timsort.

d) Im Vergleich zu den anderen vergleichsbasierten Sortieralgorithmen, die in der Vorlesung besprochen wurden, erzielt Timsort immer die bestmögliche Laufzeit für best-, average- und worst-case. In der Tabelle befindet sich kein anderer Algorithmus der in den drei Fällen die beste Laufzeit liefert. Timsort ist außerdem ein stabiler Sortieralgorithmus, sodass Elemente mit gleichem Schlüssel nicht vertauscht werden.
Aufgrund seiner Laufzeit und Stabilität eignet sich Timsort gut als Standard-Sortieralgorithmus für die Python Umgebung, da die Sortierung maximal schnell abläuft und zusätzlich die Sortierung von gleichen Schlüsseln beibehalten wird. Diese Eigenschaft ist besonders beim Sortieren nach mehreren Schlüsseln (beispielsweise Tabellen) notwendig.

Algorithmus	best	average	worst	in-place	stabil
Selection Sort	n^2	n^2	n^2	Ja	Nein
Insertion Sort	n	n^2	n^2	Ja	Ja
Bubble Sort	n	n^2	n^2	Ja	Ja
Quick Sort	$nlog(n)$	$nlog(n)$	n^2	Nein	Nein
Merge Sort	$nlog(n)$	$nlog(n)$	$nlog(n)$	Nein	Ja
Heap Sort	$nlog(n)$	$nlog(n)$	$nlog(n)$	Ja	Nein
Timsort	n	$nlog(n)$	$nlog(n)$	Nein	Ja

Tabelle 1: Vergleichsbasierte Sortieralgorithmen