

Quicksort und Pivotelementwahl

Dennis Sentler, Malte Janssen

8. November 2017

Zusammenfassung

Es wurde ein Quicksortverfahren nach dem Muster von dem Skript für Algorithmen und Datenstrukturen WS 2017 implementiert und mit mehreren Pivotwahlverfahren ergänzt. Zusätzlich wurde die Effizienz dieser Wahlverfahren getestet. Außerdem wurden Gedanken darüber gemacht, wie denn die best-, average- und worst-cases getestet werden können.

Das Quicksort-Verfahren profitiert von der richtigen Wahl des Pivotelements. Denn dieses teilt die r

1 Wahl des Pivotelements

Das Quicksort-Verfahren profitiert von der richtigen Wahl des Pivotelements. Denn dieses teilt die rekursiv zu bearbeitenden Teillisten. Desto gleichmäßiger die Teilung erfolgt, desto weniger Vergleiche müssen gemacht werden.

In einer großen Liste müssen alle Elemente unter sich verglichen werden. Teilt man die Listen hingegen perfekt auf, kann man in der Theorie den Aufwand enorm reduzieren.

2 Letztes Element als Pivotelement

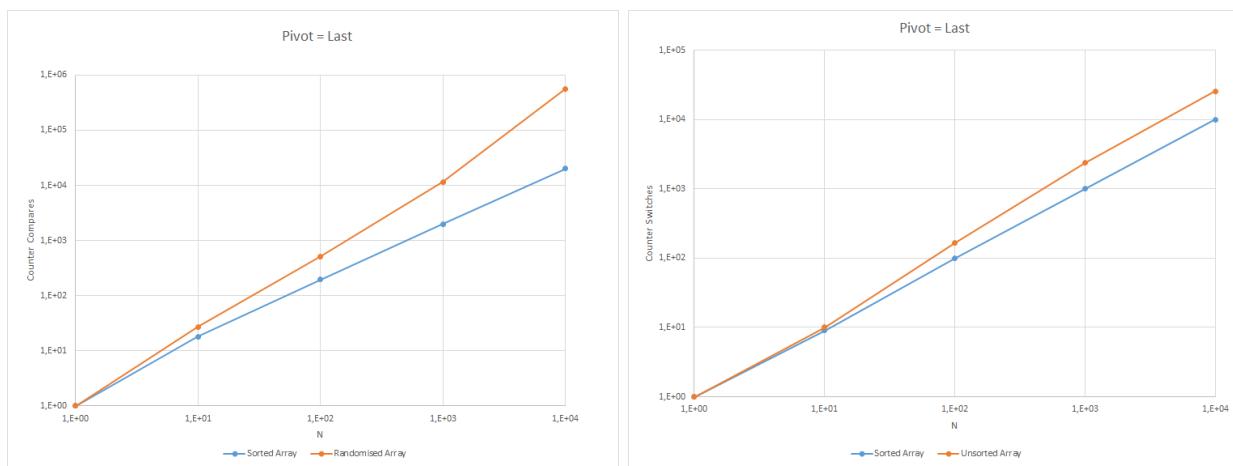
Als eine der Wahlvarianten wurde das letzte Element der Liste als Pivotelement gewählt, dies ist eine sehr einfache und schnelle Lösung. Andererseits ist die Effektivität des Quick-sortverfahrens dabei stark abhängig von den enthaltenen Elementen und der Reihenfolge.

2.1 Tabellarische Darstellung

Tabelle 1: Quicksort mit dem letzten Pivotelement				
Größe	Vergleiche		Tausch	
	Sortierte Liste	Zufällige Liste	Sortierte Liste	Zufällige Liste
1	0	0	0	0
10	18	27	9	10
100	198	513	99	165
1000	1998	11573	999	2385
10000	19998	562397	9999	25677

Entgegen der Erwartungen schnitt die vorsortierte Liste besser ab, als die, mit den zufällig generierten Inhalten. Sowohl die Anzahl bei den Vergleichen, als auch das Tauschen fiel weniger Aufwändig aus. Hiermit kann man sagen, dass die sortierte Liste für das Pivotelement auf letzten Position, den best-case darstellt. Die Zufällige Liste hingegen den aufwändigeren, worst-case.

2.2 Grafik



(a) Vergleiche zwischen Elementen

(b) Tauschen von Elementen

Abbildung 1: Quicksort mit dem letzten Pivotelement

3 Median als Pivotelement

Die zweite implementierte Methode zur Pivotwahl ist mit dem Median gelöst worden. Drei Elemente wurden dabei eingelesen, das erste, letzte, und welches sich auf mittlerer Position befindet. Diese werden nach dem zu sortierenden Attribut verglichen und der Median wird als Pivotelement gewählt. Dieses Vorgehen macht gleichmäßigere Aufteilung wahrscheinlicher, ohne die Aufwand zu erhöhen. Denn drei Elemente zu vergleichen ist ein konstanter Aufwand und hängt nicht von der Problemgröße ab.

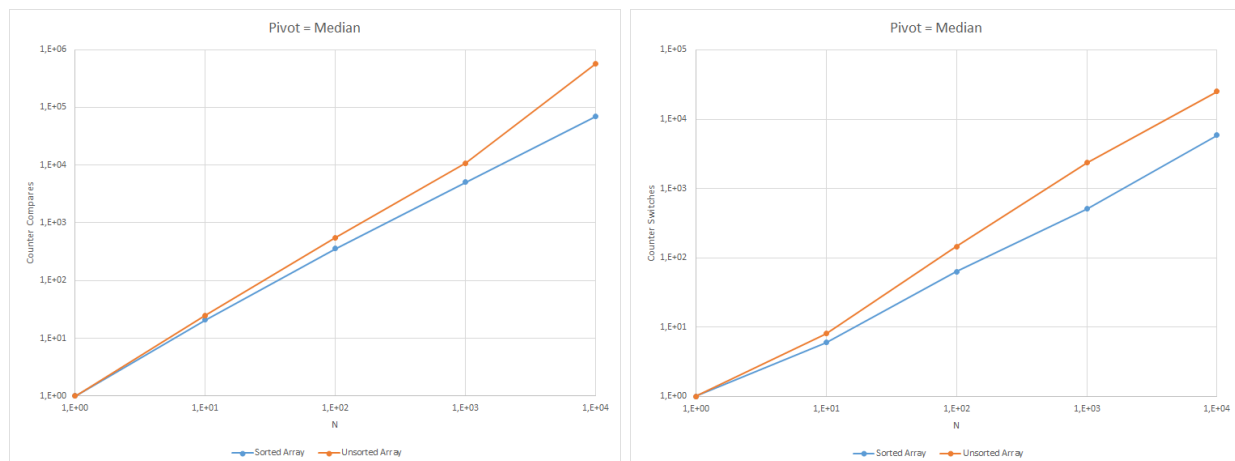
3.1 Tabellarische Darstellung

Tabelle 2: Quicksort mit dem Median-Pivotelement

Größe	Vergleiche		Tausch	
	Sortierte Liste	Zufällige Liste	Sortierte Liste	Zufällige Liste
1	0	0	0	0
10	21	25	6	8
100	360	556	63	146
1000	5048	10796	511	2306
10000	69017	561026	5904	25247

Dieses Untersuchungsergebnis, mit dem Median als Pivotelement bestätigt das Ergebnis aus der Untersuchung aus dem Punkt 2 (Seite 2). Die sortierte Liste stellt sich ebenfalls wieder als, der best-case, sowohl bei den Vergleichen, als auch bei den Tauschoperationen.

3.2 Grafik



(a) Vergleiche zwischen Elementen

(b) Tauschen von Elementen

Abbildung 2: Quicksort mit dem Median-Pivotelement