

Dokumentation zum Aufgabenblatt a06

Implementiert wurde der Quicksort-Sortieralgorithmus mit drei grundsätzlich verschiedenen Pivotsuchverfahren. Der Algorithmus ist dem im Skript angegebenen Algorithmus nachempfunden. Lediglich ein paar Erweiterungen bzw. Änderungen sind vorgenommen worden:

- Die äußere while(1) Schleife ist durch eine do-while-Schleife ersetzt worden, diese terminiert, wenn sich die Laufindizes (i und j) der beiden inneren while-Schleifen treffen.
- Die beiden inneren while-Schleifen sind um ein Abbruchkriterium (Stoppmechanismus) erweitert worden. Die erste while-Schleife darf maximal bis zum größten Index laufen, die zweite Schleife bis zum kleinsten Element. Die inneren Schleifen laufen sequentiell nacheinander ab. Die erste Schleife rennt bis ein Element gefunden wurde, welches kleiner oder gleich dem Pivotelement ist. Die zweite Schleife terminiert, wenn ein Element größer oder gleich dem Pivotelement ist.
- Die if-Anweisung nach den beiden inneren Schleifen erspart ein ggf. unnötigen Tausch, genau dann, wenn die beiden Laufindizes der inneren Schleifen (i und j) gleich sind.
- Die zusätzliche if-Anweisung nach der äußeren while-Schleife prüft ob das Element an der Position i im Array größer dem Pivotelement ist. Falls dies zutrifft, ist das jetzige Pivotelement der Liste mit den kleineren Elementen zuzuordnen. Daher werden die Positionen i und Pivot getauscht.

Implementiert wurden die im Folgenden beschriebenen Pivotsuchverfahren:

- Im ersten Verfahren wird immer das Element mit dem größten Index als Pivotelement gewählt
- Im zweiten Verfahren wird aus den Schlüsselwerten der Elemente mit dem kleinsten, dem mittleren und dem größten Index der Median als Pivotelement gewählt. Dies gilt aber nur dann, wenn alle drei Schlüsselwerte verschiedenen sind. Falls mindestens zwei der Schlüsselwerte gleich sind, wird das Element mit dem größten Indexwert als Pivotelement gewählt.
- Beim dritten Verfahren wird ein zufälliger Index bestimmt. Dieser liegt zwischen dem kleinsten und größten übergebenen Index.

Die drei Fälle Worst-, Best- und Average Case setzen sich wie folgt zusammen:

- Im Best Case wird das Pivotelement immer so gewählt, dass die beiden entstehenden Teillisten etwa gleich groß sind. In diesem Fall gilt für die asymptotische Laufzeit des Algorithmus $O(n \cdot \log(n))$.
- Für den Average Case gilt der gleiche Aufwand von $O(n \cdot \log(n))$, wie im Best Case., bei einer durchschnittlichen Länge der Teillisten von $\frac{1}{n} \sum_{i=0}^{n-1} i = \frac{n-1}{2}$
- Im Worst Case wird das Pivotelement immer so gewählt, dass es das größte oder das kleinste Element der Liste ist. Dies ist etwa der Fall, wenn als Pivotelement stets das Element am Ende der Liste gewählt wird und die zu sortierende Liste bereits sortiert ist. Die zu untersuchende Liste wird dann in jedem Rekursionsschritt nur um eins kleiner und die Zeitkomplexität wird beschrieben durch $O(n^2)$.

Die geforderte Auswertung und Darstellung der Ergebnisse in Form einer Tabelle und Diagrammen befindet sich im Anschluss an diese Dokumentation.

Martin Slowikowski

Matrikelnummer: 199 91 66

Tell Mueller-Pettenpohl

Matrikelnummer: 198 99 82

Teamname: Tugend und Laster

Quellenangaben

TIB3-AD-skript.pdf

Letzte Änderung 22.03.2011

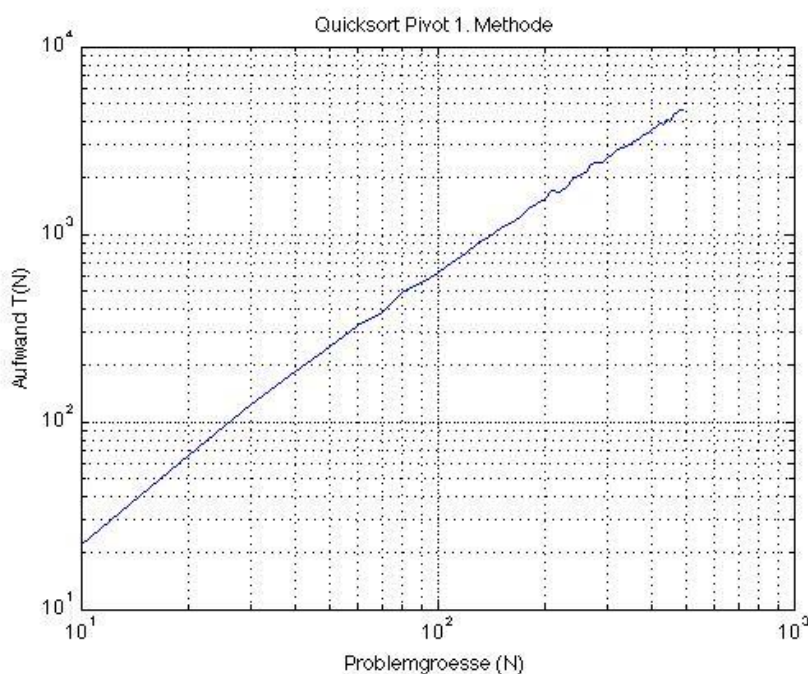
Auswertung und Darstellung

Es folgen die grafischen Auswertungen der eingebauten Zähler der einzelnen Pivotverfahren mit dem Quicksort-Algorithmus.

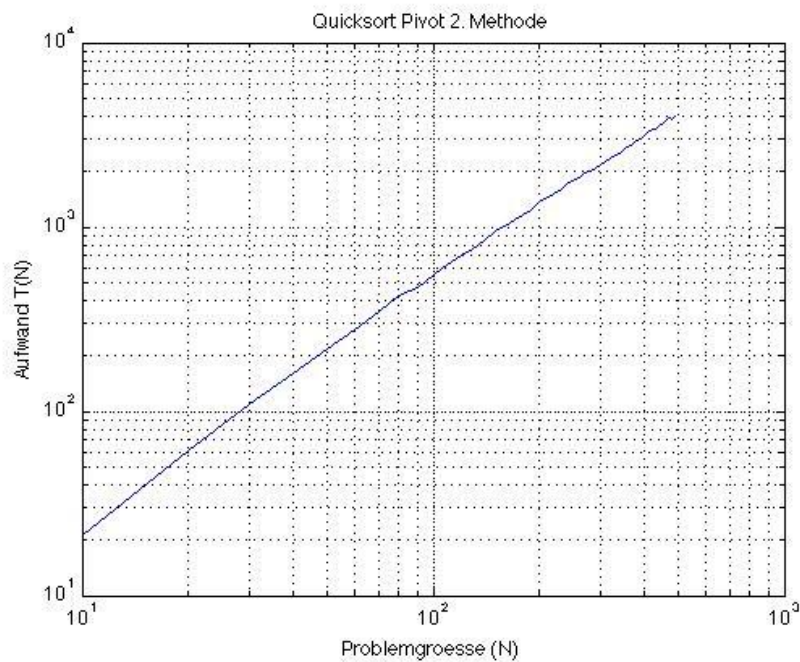
Für jedes der drei Pivotverfahren wurden die Messwerte wie folgt ermittelt:

- Die Schrittweite der Problemgröße (Arraylänge N) beträgt 10. Die Arraylänge wird bis zu einer Maximalgröße von 500 in 10er Schritten erhöht.
- Für die Anzahl der Operationen (Aufwand $T(N)$) zu jeder Problemgröße wurde der Mittelwert von 20 Iterationen mit jedem Pivotverfahren berechnet

So kann von einem durchschnittlichen Verhalten beim Sortieren ausgegangen werden.

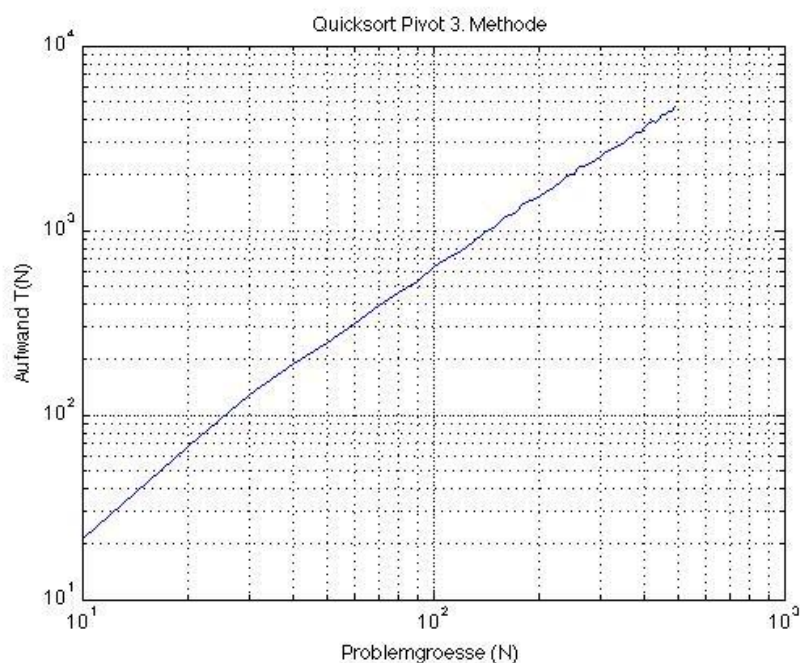


Bei der ersten Pivotmethode wird immer der Schlüsselwert des Elements mit dem größten Index gewählt. Der Aufwand bei dieser Pivotmethode beträgt $O(n \cdot \log(n))$.



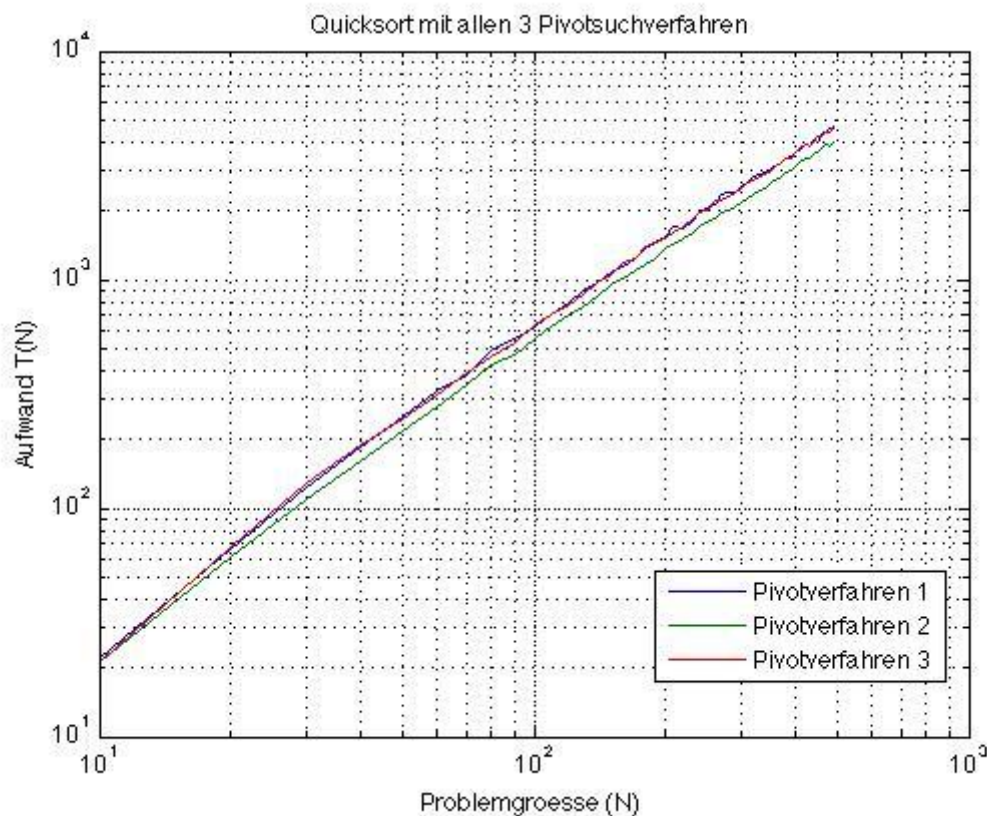
Im zweiten Verfahren wird aus den Schlüsselwerten der Elemente mit dem kleinsten, dem mittleren und dem größten Index der Median als Pivotelement gewählt. Dies gilt aber nur dann, wenn alle drei Schlüsselwerte verschiedenen sind. Falls mindestens zwei der Schlüsselwerte gleich sind, wird das Element mit dem größten Indexwert als Pivotelement gewählt.

Auch hier ist ein superlinearer Aufwand zu erkennen, $O(n \cdot \log(n))$.



Beim dritten Verfahren wird ein zufälliger Index bestimmt. Dieser liegt zwischen dem kleinsten und größten übergebenen Index.

Wie auch in den beiden vorherigen Grafiken ist auch hier ein Aufwand von $O(n \cdot \log(n))$ zu erkennen.



Betrachtet man alle drei Graphen in einem Koordinatensystem, erkennt man, dass das erste und das dritte Pivotverfahren einen nahezu gleichen Aufwand haben. Dies ist auch logisch, da in beiden Fällen zufällige Elemente gewählt werden.

Das zweite Pivotverfahren hat einen geringfügig besseren Aufwand, als die anderen beiden. Begründen kann man dies mit der Eigenschaft des Verfahrens, dass aus drei unterschiedlichen Elementen stets das günstigste (mittlere) gewählt wird.