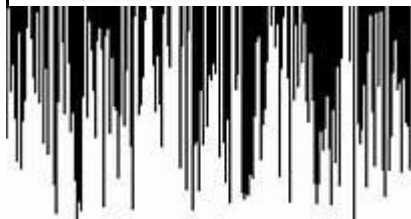
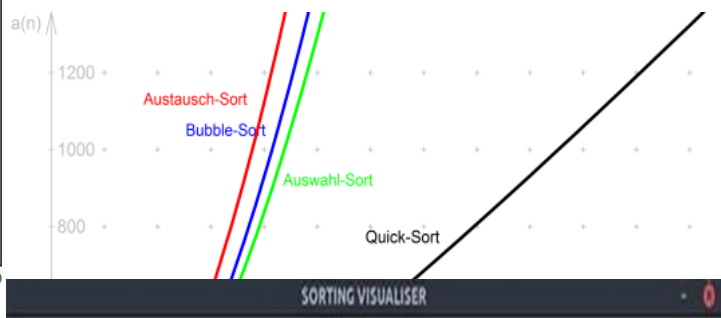


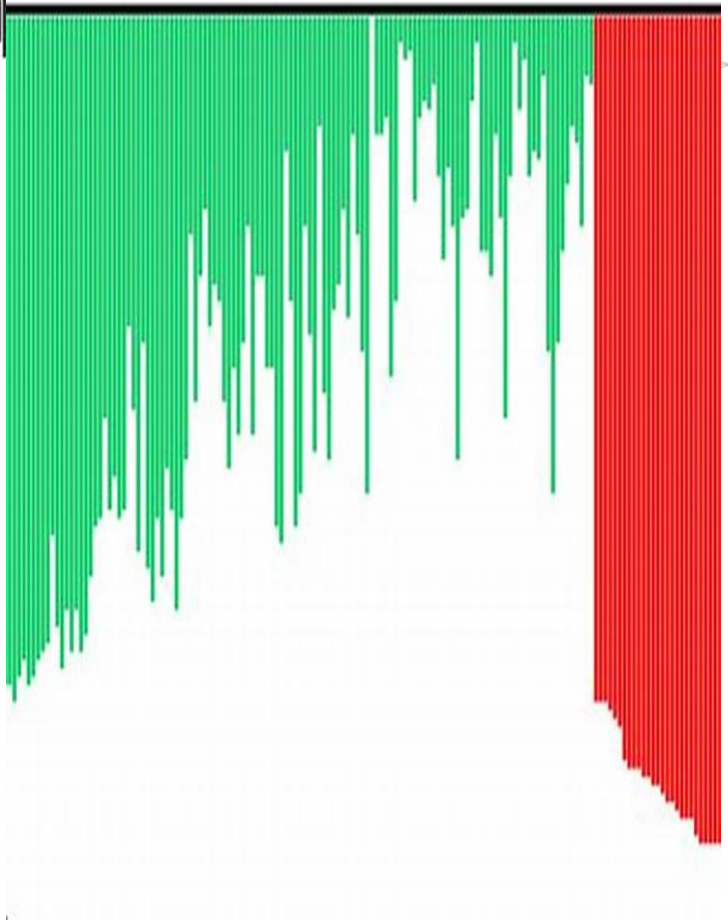
Array store operations: 67,088
Array fetch operations: 157,312



SORT: PRESS 'ENTER'
NEW ARRAY: PRESS 'R'

Running Time(sec): 37

ALGORITHM USED: HEAP SORT



Gliederung

- Einleitung
- Heranführung an Computer
- Funktionsweise
- Algorithmen im Vergleich
- Fazit
- Quellenverzeichnis
- Eidstattliche Erklärung

Heranführung an Computer

Computer sprechen ihre eigene Sprache, wo verschiedene Computer verschiedene Sprachen haben und nutzen. Dies ist eine grundlegende Einführung in Datenformate. Wenn Computer Daten speichern, nutzen sie Datenformate, Pakete und ähnliche. Allerdings kann man einem Computer nicht einfach sagen, was er machen soll. Man muss verschiedene Fälle decken, und logisch denken, um Fehler zu vermeiden. Ein Computer spricht auch kein Deutsch oder Englisch. Er spricht eine Sprache, obwohl er diese eher liest und ausführt, welche optimiert für seine Anwendungszwecke ist. Es gibt circa 700 Programmiersprachen, die für ihre Anwendungsbereiche optimiert sind.

Ein passendes Beispiel wäre „Cobol“, es wurde entworfen, und ist so minimal, sie kann höchstens 80 spalten, und 10 Zeilen haben, und passt auf eine Handgroße Lochkarte. Sie wird immer noch in Banksystemen genutzt.

Zu den bekanntesten zählen zum Beispiel **Strings**, welche eine Zeichenkette beinhalten. Man trifft auch oft den **Integer** an welcher eine ganze Zahl speichern kann, **Boolean**, welcher eine Schaltvariable darstellt, dieser ist entweder wahr, oder falsch sein kann. Es gibt auch den **Float**, dieser wird genutzt, um Gleitkommazahlen darzustellen. Aber das Datenformat, dass heute am wichtigsten ist, ist der sogenannte **Array** (eng. → de. „Anordnung/Feld“).

Man kann einen Array leicht erklären, wenn man sagt, dass sie eine Liste darstellen, in welchem man verschiedene Datenformate einordnen, kopieren, sortieren und anderes machen kann. Es ist eine weitverbreitete Art um mit großen Daten zu arbeiten. Viele, wenn nicht alle Programme arbeiten mit Arrays. Man kann in diesen die zuvor genannten Formate speichern: **Integer**, **Float**, **String**, **Boolean**, man kann sogar Arrays in Arrays speichern.

Nun. Es ist wichtig, dass solche Arrays geordnet sind, zum Beispiel nach Länge, Alphabetisch oder anderen Kriterien. Um solche Sortierprozesse durchzuführen, nutzt man die sogenannten Sortieralgorithmen. Heute beschäftige ich mich mit dem Quicksort. Und erkläre die Funktionsweise. Ein grundlegender Algorithmus.

Die Funktionsweise

Vorweg sei gesagt, dass Quicksort durch den englischen Mathematiker und Informatiker „Sir Charles Anthony Richard Hoare“. Der Grundgedanke bestand darin, das Gesamtfeld in zwei Teilfelder zu zerlegen mit Hilfe des Pivotelements, auf welches ich gleich noch eingehe.

Ganz grundlegend gehört der Quicksort zu den rekursiven und nicht instabilen Sortieralgorithmen. Der Quicksort arbeitet nach dem sogenannten „Divide-and-Conquer“ Prinzip, was soviel heißt, wie Teile-und-herrsche.

Als erstes ist es wichtig, zu verstehen, was das Pivotelement ist. Dies heißt übersetzt Angelpunkt, dies gilt als Ausgangspunkt, wo der Algorithmus mit seiner eigentlichen Arbeit erst beginnt. Um das Pivotelement zu bestimmen wählt, was je nach Programmiersprache, System oder Konfiguration variiert, ist entweder das erste, letzte, das in der Mitte oder sogar ein zufällig ausgewähltes Element ein.

Als nächstes wird das Pivotelement in die Mitte des Arrays gesetzt, und dann werden die restlichen Werte sortiert. Sie werden nach links sortiert, wenn sie kleiner als das aktuelle Vergleichselement sind, dementsprechend auch nach, rechts, wenn das Element größer ist. Dabei muss man die Elemente immer der ursprünglichen Reihenfolge nach von links nach rechts in ihrem Bereich einordnen.

Es gibt durchaus auch verschiedene Varianten und Abwandlungen des Quicksorts. Allerdings unterscheiden sich die meisten Varianten unterscheiden sich allerdings fast ausschließlich in der Handhabung des Pivotelements. Es wird auf die Aufgaben, Anforderungen und gedachten Nutzen optimiert, um den Zweck möglichst gut zu erfüllen. Der Hauptanwendungsbereich des Quicksort liegt dabei, Datenbanken zu verwalten, Textverarbeitungsprogramme zu strukturieren, oder Compiler zu

Algorithmen im Vergleich

Hier vergleichen wir den Mergesort mit dem Quicksort, vorab: Mergesort und Quicksort basieren beide auf dem Teilen-und-Einnehmen Prinzip, was bedeutet, dass diese Algorithmen gar nicht so fern von einander sind, sondern viel gemeinsam haben.

Anders als der Quicksort, ist der Mergesort ein stabiler Algorithmus. Der Mergesort schneidet im Fenster Performance bei größeren und komplexeren Datenpaketen, wesentlich besser ab. Der Mergesort ist dagegen kein In-place-Algorithmus. Das heißt, dass dieser Algorithmus nicht ausschließlich stabile Sortierungen liefert. Das schont den Speicher weniger, wenn er nicht bei den richtigen Anwendungszwecken zum Einsatz kommt.

Ein weiterer Konkurrent des Quicksort ist der Bubblesort. Aber wenn man alles Zusammenfasst, ist der Quicksort effizienter als der Bubblesort, da dieser den gegebenen Array in viele einzelne Arrays zerlegt, indem er immer Zweierpaare aus dem Array nimmt, und diese vergleicht und dann in der korrekten Reihenfolge wieder zusammensetzt. Dies spielt der Quicksort aus, indem es das Pivotelement nutzt, um Performance beizubehalten.

Man kann abschließend sagen, dass zwar jeder Algorithmus, um das Problem von Sortierungen in Arrays zu lösen, seine Vor- und Nachteile basierend auf den genauen Anwendungsbereich hat, aber man kann mit Sicherheit sagen, dass die heutige Computertechnik rasant voranschreitet, und somit der Quicksort durch seine Funktionsweise ein Spitzenreiter der Algorithmen ist. Und nicht ohne Grund, einer der führenden Algorithmen unserer Zeit ist.

Fazit

Einleitung

Ich finde das Thema Algorithmen an sich schon extrem interessant, und möchte mein erlangtes Wissen zu diesem Thema, einfach teilen. In dieser Ausarbeitung erkläre ich wesentliche Punkte die zum Thema Sortieralgorithmen relevant und gut zu wissen sind. Ich möchte zudem erklären, wie Computerdenken, da dies ein sehr komplexes Thema, mit einer wahnsinnig steilen Lernkurve ist, gehe ich auf vereinfachte Konzepte ein, um mein Denken verständlicher zu gestalten, und besser erläutern zu können. Ich hoffe man kann meinen Erklärungen erfolgreich folgen und verstehen.

Eidstattliche Erklärung

Am Schluss der Arbeit hat der Verfasser eine Versicherung mit folgendem Wortlaut abzugeben: „Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und bei ihrer Adertilgung keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die anderen Werken dem Wortlaut oder Sinn nach entnommen sind, habe ich unter Angabe der Quelle der Entleihung kenntlich gemacht.“

Quellen

- Baeldung on Computer Science
- happycoders.eu
- Medium
- Computer Science Stack Exchange
- interviewkickstart.com