

# Visualisierung – Project – Report

Elif Bilgin

Franz Brandl

Universität Wien

## ABSTRACT

Die Lehrveranstaltung „Algorithmen und Datenstrukturen“ ermöglicht, dass die Studenten ihre Implementierungen graphisch vergleichen. In dieser Paper präsentieren wir eine interaktive Lösung, damit die Studenten ein klares Verständnis haben können, wie schnell oder langsam ihre Implementierungen sind.

## 1 MOTIVATION

Die Studenten von Universität Wien, welche an der Lehrveranstaltung „Algorithmen und Datenstrukturen“ teilnehmen, müssen verschiedene Datenstrukturen implementieren, um diese Lehrveranstaltung zu schaffen. Nachdem sie eine laufende Implementierung abgegeben haben, haben sie die Möglichkeit, ihre Implementierungen mit anderen Studenten und auch mit der Referenzimplementierung zu vergleichen. Die aktuelle Visualisierung hat aber keine Interaktivität, deswegen muss man es von vielen verschiedenen Diagrammen analysieren, deswegen ist es nicht so einfach für die Studenten die Ergebnisse zu vergleichen. Wir wollten deswegen eine neue Visualisierung zu erstellen, damit die Ergebnisse für Studenten auf einen Blick erkennbar sind.

### 1.1 EXAMPLE SCENARIO

Ein Student, wer als Thema „Double Hashing“ hat, möchte wissen, ob er eine gute Algorithm verwendet hat, deswegen will er seine Implementierung mit Referenzimplementierung als auch mit den Implementierungen anderen Studenten, die auch als Thema „Double Hashing“ haben vergleichen. Als erstes bekommt er eine Übersicht in der die Methoden pro jeder „size“ verhältnismäßig mit der Referenz verglichen werden. Will er sich dann eine Methode genauer ansehen kann er diese auswählen. Dann bekommt er eine Graph, welche genau bekannt gibt, wie schnell seine Implementierung im Vergleich zum Referenzimplementierung und durchschnittliche Performanz von Implementierungen den Studenten ist. Er hat die Möglichkeit jede einzigen Methode zu vergleichen. Somit weist er, ob seine Implementierung schnell genug ist, oder er seine Implementierung verbessern soll. Wenn er sich entscheidet, dass er seine Implementierung verbessern soll, dann weiss er genau, welche Methode oder Methoden er verbessern soll.

## 3 Approach

Als wir erst unsere Projekt ausgedacht haben, war es ganz klar, dass wir es ermöglichen sollen, dass ein Student seine Implementierung mit Referenzimplementierung vergleicht. Aber es war nicht so klar, mit welchen Studenten wir seine Implementierung vergleichen sollen. Erst haben wir gedacht, dass wir es mit schlechteste 5 Studenten, beste 5 Studenten und Durchschnitt von alle Studenten vergleichen sollen.

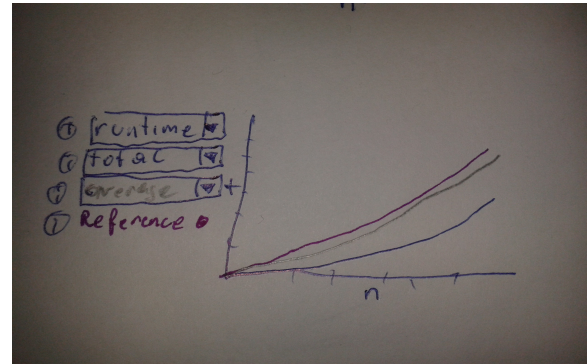


Figure 1: Proposal Design

Dann haben wir gedacht, dass man eigentlich es mit schlechteste 5 nicht vergleichen würde, weil das Ziel es immer zu verbessern ist. Man würde es aber mit dem Durchschnitt vergleichen, weil jeder würde gerne wissen wollen, ob er besser oder schlechter als Mehrheit ist. Vergleichung mit den besten 5 war keine schlechte Idee, aber wir konnten nicht eine optimale Lösung finden, um die beste 5 festzustellen. Nachdem wir festgestellt haben, was wir genau machen wollen, mussten wir uns entscheiden, welche Tool wir benutzen sollen. Erst haben wir ein Paar Tools gefunden, welche sehr einfach zu benutzen sind, aber am Ende haben wir uns entschieden, dass wir bei unserem Projekt mit d3 beschäftigt sein wollen, weil wir vorher schon mit d3 ein bisschen beschäftigt waren und weil wir unser Kenntnisse zum d3 verbessern wollten.

## 4 IMPLEMENTATION

Unser Data ist die XML-Dateien, welche die Informationen von den Implementierungen von Studenten und die Informationen von Referenzimplementierungen von jedem Thema beinhalten. Jede XML-Datei erfasst die Performanzinformationen von passenden Student oder Referenz für jede Methode bei jeden „n“. Wenn ein Student seine Implementierung testen will, laden wir einfach die betreffende Informationen von dem Student und auch von Referenz. Für den Durchschnitt und die besten Werte haben wir csv-Files für jedes Thema erstellt. Das haben wir in C++ mit Hilfe von rapidxml umgesetzt.

## 5 Results

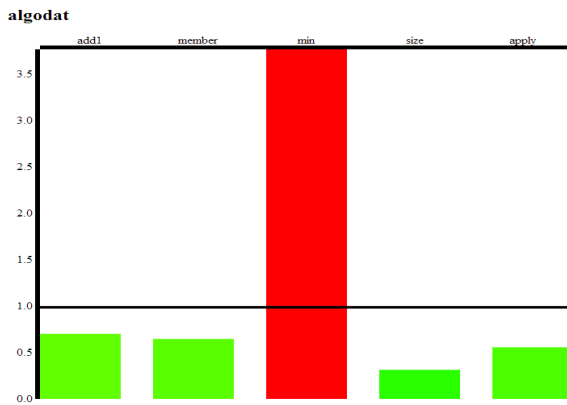


Figure 2: Overview(alle Methoden gut außer min)

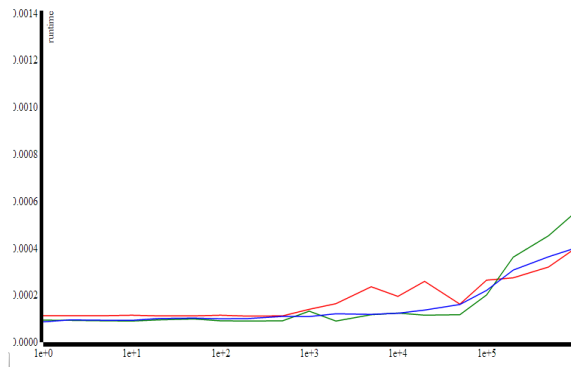


Figure 3: Methode ausgewählt(grün Referenz, rot Student, blau Durchschnitt)

## 6 Discussion

Die Hauptsache bei unserem Projekt ist, dass die Benutzer eine klare Overview bekommt, und sich mit den anderen vergleicht. Und damit sie es möglich einfach schaffen können, müssten wir eine interaktive Plattform erzeugen. Es gab aber in unserem Data mehrere Informationen, mit welchem man arbeiten und mehrere Sachen machen kann, z.B. wir wollten noch die Allocated-Memory vergleichen usw. aber weil wir bis jetzt keine Erfahrung mit der Visualisierung von Daten haben, gab es für uns vieles zu lernen, und es hat viele Zeit gekostet, diese korrekt zu implementieren. Wie man in unserem Proposal sehen kann, wollten wir eigentlich mehrere Sachen machen, aber wir brauchten noch Zeit, um diese auch zu schaffen.

### 6.1 Lessons Learned

Elif : Was ich durch diese Aufgabe gelernt habe ist, dass man entweder am Anfang ganz genau wissen soll, wie man es Schritt zu Schritt implementieren wird oder man die Zeit sehr gut managen kann, wenn man beschränkte Zeit hat, und viel neues lernen muss. Ich hatte bis jetzt nie mit XML-Dateien gearbeitet, mit Java-Script und mit d3 auch nicht so oft, deswegen könnte ich sagen, dass ich während der Ablauf des Projekts viele neue Sachen gelernt habe.

Franz : Ich habe bei dieser Aufgabe viel über Javascript, HTML und XML gelernt. Großteil meiner Aufgabe war der Umgang mit den Daten. Um mit XML-Daten zu arbeiten habe ich mehrere Libraries versucht und viel über diese gelernt(tinyxml, rapidxml).

Da ich vorher kaum etwas mit html oder javascript zu tun hatte habe ich auch heir vieles neues gelernt. Natürlich habe ich auch viel über Visualisierung gelernt, das Problem des Vergleichens bei Kurven, oder die Codierung von Information über mehrere Kanäle sind ein paar mit denen ich mich bei der Arbeit an dem Projekt näher beschäftigt habe.