

Mathematisches Praktikum¹

im Studiengang Mathematik (Bachelor of Science).

16. August 2010 (SS 2010, RWTH Aachen)

Florian Weingarten <flo@hackvalue.de>

Prüfungsdauer: etwa 25 Minuten

Prüfungsnote: 1.0

Achtung: Hierbei handelt es sich *nicht* um ein offizielles Prüfungsprotokoll der RWTH sondern um ein privates Gedächtnisprotokoll, das ca. eine Stunde nach der Prüfung angefertigt wurde. Ich habe mit Sicherheit einige Sachen vergessen und gebe natürlich keine Garantie auf Korrektheit.

Codeschnipsel erklären

- `include`-Statements erklären, was ist `iostream`
- `using namespace` Zeug erklären, wozu braucht man Namespaces?
- Programm war eine `for`-schleife von $i = 0$ bis $i < 10$ (und $i++$) mit Ausgabe von i in jedem Schritt, aber zusätzlich noch mit $i = i \cdot 5$ im Rumpf (Ausgabe: 0, 5, 30)
- Was ist `main`? Warum hat `main` einen Rückgabewert vom Typ `int`?

Programmablaufpläne

- Was ist das? Welche Symbole gibt es? Wozu braucht man das?
- Gegeben eine natürliche Zahl n . Zeichnen Sie den Programmablaufplan zu einem Algorithmus, der die Quersumme von n bestimmt. (Mein Ansatz: sukzessiv Rest nach Division durch 10.)

Potenzmethode

- Was ist das? Problem beschreiben, Verfahren beschreiben
- Wozu normiert man in jedem Schritt? (Overflow vermeiden)
- Warum darf man normieren? (linearer Raum, Vielfache von Eigenvektoren sind selbst Eigenvektoren)
- Nach welcher Norm normiert man (∞ -Vektornorm)
- Bestimmt man damit den Eigenvektor oder den Eigenwert? (Eigenvektor)
- Wie kriegt man den Eigenwert? (klar, Definition)

Klassen

- Wozu braucht man hier (Potenzmethode) Klassen? (schönere Notation, Kapselung, etc.)
- Was ist besonders nützlich in diesem Kontext? (Überladung der arithmetischen Operatoren für z.B. Skalarprodukt, Matrix-Vektor-Produkt, etc.)
- Syntax einer Klassendefinition grob aufschreiben
- Unterschied `private`, `public` und `protected`
- Was ist ein Konstruktor/Destruktor? **Muss** man das definieren? Wie definiert man das (Syntax)?
- Wann wird Konstruktor aufgerufen? Wann Destruktor? (`delete` oder Ende von Scope)
- Syntax von abgeleiteten Klassen (Vererbung) grob erklären
- Vektor vom Typ `double`. Jetzt will man `int`. Was tut man? (Templates benutzen)
- Warum sind Templates schön? In welchem Moment wird aus einem Template eine Klasse erzeugt? (geht zur Compilezeit)

¹<http://www.igpm.rwth-aachen.de/node/620>

Compiler

- Wird der Compiler durch Templates langsamer? (dämliche Frage. Meine Antwort war „prinzipiell ja, falls das Template mit mehreren verschiedenen Typen vorkommt, weil dann vermutlich mehrere unterschiedliche Klassen vom Compiler angelegt werden“)
- Wie kompiliert man ein Programm? (z.B. mit `g++ datei.cpp -o datei`)
- Was sind Kommandozeilenparameter? `./test 1 2 3`, was ist `argc`? Was ist `argv[0]`?

Tabu Suche

- Was ist das? (Heuristik für binäres Rucksackproblem)
- Was heißt binär? (jeder Gegenstand keinmal oder einmal)
- Warum Heuristik? (NP-vollständig)
- Was heißt NP-vollständig? (in NP und mindestens so schwer wie jedes andere NP Problem - Prüfer verdreht die Augen. Er mag wohl keine Informatiker)
- Und jetzt nochmal in kurz und knapp? (Man geht davon aus, dass das nicht effizient exakt lösbar ist, deswegen approximiert man)
- Was heißt effizient? (polynomiell in der Eingabe)
- Idee von Tabu Suche erklären (Tabuliste als FIFO-Queue, Länge der Queue heißt Tabudauer, etc.)
- Wie gut ist das Verfahren? (weiß man nicht, gibt kaum bewiesene Aussagen, aber scheint in der Praxis halbwegs gut zu funktionieren)
- Wieviele Iterationen macht man? Wie lang ist die Tabu Liste? (beides frei wählbar und beide Parameter beeinflussen den Algorithmus sehr stark)

Fazit

Für jemanden, der schonmal ein paar C++ Quelltexte gelesen hat (das reicht wirklich, ich habe eigentlich fast keine Erfahrung mit C++) war die Prüfung sehr einfach. Vor den dämlichen Programmablaufplänen hat mich ein Kommilitone zum Glück vorher gewarnt, so dass ich mir das nochmal kurz angeguckt habe. Das war das einzige was sonst etwas unerwartet gewesen wäre. Zeit ging schnell rum. Alles sehr locker. Geprüft wurde ich von zwei sehr jungen Assistenten (oder Hiwis?), die ich beide noch nie vorher gesehen habe.