Westfälische Hochschule

Prof. Dr. Manfred Meyer
Fachbereich 6 / Bocholt
Professur BWL / Wirtschaftsinformatik

Algorithmen und Datenstrukturen
(Studiengang Wirtschaftsinformatik 3. Semester)

Aufgabenstellung für das Praktikum am 12.10.2023

Vorbemerkung:

In der heutigen Vorlesung haben Sie am Beispiel des „Problems der falschen Münze“ die Erfahrung gemacht, dass es für ein gegebenes Problem verschiedene Lösungsansätze (Algorithmen) geben kann, die alle zu einer Lösung für das Problem führen (Stichwort: Terminierung und Korrektheit), deren Aufwand sich aber bei anwachsenden Problemgrößen ganz unterschiedlich entwickelt.

Im heutigen Praktikum werden Sie sich in Ihren Kleingruppen mit zwei weiteren Problemen beschäftigen, für die Sie jeweils einen einfachen („naiven“) Algorithmus bereits vorgegeben bekommen. Ihre Aufgabe besteht nun darin, diese Algorithmen zu verstehen, zu analysieren, zunächst schrittweise zu verbessern und diese Verbesserungen durch Laufzeitmessungen zu dokumentieren. Ergänzend sollten Sie auch nach weiteren bekannten Verbesserungen im Web recherchieren, diese ebenfalls implementieren und hinsichtlich ihrer Laufzeit untersuchen (messen).

Wahrscheinlich werden Sie in Ihrer Gruppe die Aufgabe 2 im Rahmen des Praktikums nicht vollständig bearbeiten können, so dass Sie hieran einzeln oder in der Gruppe gerne bis zur nächsten Vorlesung weiterarbeiten können, in der ich auf die Ergebnisse nochmal eingehen werde.

Ich wünsche Ihnen „gute Unterhaltung“ – im Sinne einer Diskussion und gemeinsamen Recherche – und viel Erfolg!

Aufgabe 1:

Gesucht ist ein Java-Programm zur Berechnung aller Primzahlen zwischen 1 und einer einzugebenden Obergrenze n. Dabei soll der zum Einsatz kommende Algorithmus schrittweise so verbessert werden, dass die benötigte Zeitspanne zum Finden der Primzahlen sukzessive verringert wird.

In Moodle finden Sie als Ausgangslage für Ihr Programm eine Vorgabe mit folgender Funktion, die einen doch recht naiven Algorithmus bereitstellt. Sie liefert eine Liste, die die Primzahlen bis zur jeweiligen Obergrenze enthält.

```
private static List<Long> primzahlenTotalNaiv (long obergrenze)
{
	List<Long> primzahlen=new ArrayList<Long>();
	boolean istPrimzahl;
	long zuPruefen=2;

	while (zuPruefen<obergrenze)
	{
		istPrimzahl=true;
		for (int teiler=2; teiler<zuPruefen; teiler++)
		{
			if (zuPruefen%teiler==0)
			{
				istPrimzahl=false;
			}
		}

		if (istPrimzahl)
		{
			primzahlen.add(zuPruefen);
		}

		zuPruefen++;
	}
	return primzahlen;
}
```

Testen Sie diesen Algorithmus, indem Sie ihn die Primzahlen bis zur Obergrenze 200.000 berechnen lassen. Was können Sie feststellen?

Diskutieren Sie in Ihrer Gruppe mögliche Optimierungen und versuchen Sie den Algorithmus zu verbessern, indem Sie die Vorgabe aus Moodle entsprechend modifizieren bzw. ergänzen.

Erweitern Sie dann das Hauptprogramm in der Form, dass die jeweils benötigte Zeit zur Berechnung der Primzahlen gemessen wird, so dass Sie die verschiedenen Algorithmen testen können.

Recherchieren Sie anschließend im Internet, welche weiteren Ansätze es zur Bestimmung von Primzahlen gibt, implementieren Sie diese und vergleichen Sie deren Leistungsfähigkeit ebenfalls mit Ihren anderen Algorithmen.

Aufgabe 2:

Gesucht ist jetzt ein Java-Programm zur Bestimmung des größten gemeinsamen Teilers (ggT) zweier natürlicher Zahlen, die über die Tastatur eingegeben werden. Dabei soll der zum Einsatz kommende Algorithmus auch wieder schrittweise so verbessert werden, dass die benötigte Zeitspanne zum Finden des ggT sukzessive verringert wird.

In Moodle finden Sie als Ausgangslage für Ihr Programm eine Vorgabe mit folgender Funktion, die einen doch recht naiven Algorithmus bereitstellt.

```
private static long ggtNaiv (long zahl1, long zahl2)
{
	long min=Math.min(zahl1, zahl2);

	for (long i=min; i>=2; i--)
	{
		if (zahl1%i==0&&zahl2%i==0)
		{
			return i;
		}
	}
	return 1;
}
```

Testen Sie diesen Algorithmus, indem Sie ihn u.a. den ggT folgender Zahlenpaare ermitteln lassen:

Zahl 1	Zahl 2
2	4.294.967.296
85	6.843.886.826
7.475.673.456	7.475.673.490

Was können Sie feststellen?

Neben diesem Verfahren, kann der ggT auch mit Hilfe der „**Primfaktorenzerlegung**“ bzw. der „**Teilermengenbestimmung“** ermittelt werden. Erarbeiten Sie in Ihrer Gruppe mögliche Algorithmen, die diese Verfahren nutzen und implementieren Sie die entsprechenden Funktionen.

Ergänzen Sie dann das Hauptprogramm in der Form, dass die jeweils benötigte Zeit zur Ermittlung des ggT gemessen wird, so dass Sie die verschiedenen Algorithmen vergleichen können.

Recherchieren Sie im Internet, welche weiteren Ansätze es zur Berechnung des ggT gibt, implementieren Sie diese und vergleichen Sie auch deren Leistungsfähigkeit mit Ihren anderen Algorithmen.

Viel Spaß und Erfolg bei der Bearbeitung der Aufgaben!