

6. Übung zu Höhere Algorithmik II

Aufgabe (20 Punkte)

Implementieren Sie die Ellipsoid-Methode in drei Teilen.

1. Das LSI-Entscheidungs-Problem: Hat ein lineares Programm mit strikten Ungleichungen eine zulässige Lösung?
2. Lösung abschätzen: Durch mehrfachen Aufruf des ersten Teils in einer binären Suchen die Lösung auf einen sehr kleinen Bereich einschränken.
3. Die Basis bestimmen (optional): Wieder durch mehrfachen Aufruf des ersten Teils die Basis bestimmen und damit den exakten Wert aller Variablen.

Die Teile bauen in dieser Reihenfolge aufeinander auf und sollten daher auch in dieser Reihenfolge implementiert werden. Der dritte Teil ist optional und bringt **extra Punkte**.

Hinweise zur Programmierung:

- Das Programm ist soweit zu kommentieren, wie es der Übersichtlichkeit dient. Also z.B. kurze Beschreibungen, was eine Funktion oder ein Programmteil tut.
- Es ist davon auszugehen, dass die Eingabedateien in Ordnung sind. Es müssen also keine Fehler in der Eingabe abgefangen werden.
- Die Eigenschaften der übergebenen Daten sind zwecks leichter Verarbeitung *public* aber auch *final* um zumindest einen Kompromiss zur sauberen Objekt-Orientierung zu haben.
- Das eingelesene lineare Programm muss erst in eine eigene Datenstruktur (z.B. eine Matrix) überführt werden.
- Der Parser aus Übung 3 bildet wieder die Grundlage. Es können Teile der eigenen oder der Muster-Lösung verwendet werden.
- Für die Ellipsoid-Methode ist das Rechnen mit hoher Genauigkeit notwendig. Benutzen Sie daher die Klasse `java.math.BigDecimal`. Auf der Homepage ist ein Link zu einer Implementierung der Quadratwurzel für diesen Typ. Sie können aber auch andere verwenden.
- Der Parameter L , der in den Formeln zur Ellipsoid-Methode vorkommt, bezeichnet die Größe der gesamten Eingabe in Bits. Also unter Anderem die Summe der Längen der einzelnen Werte der Matrix A in Bits.

Abgabe: bis 02.06.2008
(per Mail)