Dr. Andreas Borchert
Michael Wiedemann
Abteilung Angewandte Informationsverarbeitung

25. Mai 2004
Blatt 4

Allgemeine Informatik II (SS 2004)

Abgabetermin: 3. Juni 2004

4 Kindergarten-Puzzle – 15 Punkte

Nach der Theorie nun wieder zurück zur Praxis ;-). Wir wollen ein wenig puzzeln, allerdings auf einer ganz niedrigen Stufe.

Ihr habt ein quadratischen Holzrahmen der Innen-Seitenlänge k vor Euch, sowie entsprechend genau k rechteckige Holz-Klötzchen, die Ihr nun so legen solltet, dass alle Klötzchen innerhalb des Rahmens Platz finden. Dabei könnt Ihr die Klötzchen nicht aufeinander, sondern nur nebeneinander legen. Außerdem dürfen die Klötzchen nicht diagonal in den Rahmen gelegt werden, sondern nur horizontal oder vertikal, das ganze soll ja auch schön passen.

Man könnte das ganze auch mathematisch formulieren, schließlich sind wir hier an einer Hochschule ;-) :
Gegeben sei ein Quadrat der Seitenlänge k sowie genau k Rechtecke mit Größen $m_1 \times n_1, \ldots, m_k \times n_k$. Zu finden ist eine Plazierung der k Rechtecke in dem Quadrat, ohne dass sich die Rechtecke überlappen. Außerdem soll die Plazierung innerhalb des durch das Quadrats definierten $k \times k$ Rasters liegen.

So etwas lässt sich vortrefflich durch Backtracking bewerkstelligen. Dabei ist folgendes zu beachten:

- Eine kleine Argumentverarbeitung ist einzubauen, so dass die Daten entweder aus einer Datei oder von der Standardeingabe kommen. Zu den Daten selbst: es werden einfach nacheinander alle maßgeblichen Daten eingelesen: zuerst die Seitenlänge des Quadrats k, dann m_1, dann n_1, usw. Dabei sollten zumindest offensichtliche Fehleingaben abgefangen werden, das heißt, zuviele oder zuwenige Rechtecke, zu große Rechtecke sowie nichtpositive Seitenlängen. Einige Testdaten sowie Beispielaufrufe finden sich in den Beispielen zu diesem Übungsblatt.
- Um die Arbeitsweise gut verfolgen zu können, bietet sich eine Ausgabe des Quadrats und der momentanen Lage der gelegten Rechtecke an. Zu Testzwecken können alle Interimsfelder ausgegeben werden, später reicht auch das Ergebnis, wenn eines existiert.
- Überlegt Euch gut, welche Datenstrukturen Ihr benötigt. Zum einen müssen die eingelesenen Daten gehalten werden, zum anderen braucht Ihr ein quadratisches Feld, um die Ausgabe zu realisieren.
- Prinzipiell solltet Ihr schrittweise vorgehen: zunächst die Argumentverarbeitung und das Einlesen der Daten sowie die Fehlerbehandlung erledigen. Als nächstes kommt der Aufbau des Quadrats und dessen Ausgabe. Zum Abschluss kommt dann das eigentliche Backtracking.
- Wenn Ihr Euch die Beispielausgaben genauer anschaut, werdet Ihr feststellen, dass meine Lösung nicht unbedingt sehr effizient arbeitet. Fallen Euch Optimierungsmöglichkeiten ein? Berichtet Eurem Tutor davon!

Viel Erfolg!