Prof. Dr. Franz Schweiggert
Norbert Heidenbluth

Allgemeine Informatik II (SS 2006)

für

(Wirtschafts-)Mathematiker/Physiker und E-Techniker

Abgabetermin: 16. Mai 2006

Vorbemerkung: In der nächsten Woche tauschen wir Übungen und Vorlesung, d.h. die Übungen finden bereits am Dienstag (16.05.2006) statt. Damit ist der Abgabetermin dieses Blattes bereits der 16.05.2006 (per E-Mail an die Tutoren)! Die Tutorien finden hingegen zur gewohnten Zeit statt.

In diesem Übungsblatt geht es darum, einen Algorithmus zu finden, der nach dem Prinzip Divide et Impera *arbeitet. Das Gesamtproblem soll also rekursiv in zwei kleine Probleme zerlegt werden, und die Gesamtlösung setzt sich nachher aus den Teillösungen zusammen.*

Aufgabe 7: Die längste Theke der Welt (10 Punkte)

Jetzt – wo die Biergärten wieder locken – wird es Zeit, das Übungsblatt mal unter das Motto einer virtuellen Kneipentour zu stellen. Und dazu nehmen wir uns am Besten die Stadt Düsseldorf als Beispiel, denn dort gibt es die längste Theke der Welt[1].

Stellen wir uns für unser Blatt also eine Reihe von n benachbarten Biergärten (in Düsseldorf: "Kneipen") vor, in denen Sie immer auf eine Gruppe von Leuten treffen, die Sie kennen und mit denen Sie ein wenig bei "geistigen Getränken" verweilen. Stellen wir uns weiter vor, bei manchen Biergärten würden Sie von Ihren Bekannten eingeladen, in den anderen müssen Sie die Bekannten einladen, wobei die Beträge, die Sie jeweils ausgeben müssen (negativ) bzw. spendiert bekommen (positiv), im voraus für jede Kneipe feststehen.

Nehmen wir als Beispiel die folgende Tabelle:

Kneipe Nr.	1	2	3	4	5	6	7	8	9	10
Betrag in Euro	3,10	-4,10	5,90	2,60	-5,30	5,80	9,70	-9,30	-2,30	8,40

[1] Und die Stadt ist abgesehen davon auch meine Heimat!

Als (Wahl-)Schwabe sind sie nun daran interessiert, eine Kneipentour dergestalt zu planen, daß Sie am Ende des Abends so viel wie möglich spendiert bekommen haben. Wobei für Ihre Kneipentour eine Einschränkung gilt: Sie dürfen **nur nebeneinander liegende Biergärten** (d.h. keine Auslassungen) und jeden Biergarten nur **(höchstens) einmal** besuchen.

Mit anderen Worten: Sie suchen also ein Paar von Biergärten (b_s und b_z), so daß der Saldo aus dem Besuch dieser und aller dazwischen liegenden Biergärten möglichst hoch ist. Falls es nur Biergärten geben sollte, in denen Sie etwas spendieren müssen, so bleiben Sie zu Hause (d.h. der Saldo ist 0).

Beispiel: Machen wir das Problem nochmals kurz an einem Beispiel klar. Angenommen, Sie starten in Biergarten 1 und enden in Biergarten 4, dann haben Sie einen Saldo von $3,10 - 4,10 + 5,90 + 2,60 = 7,50$. Besuchen Sie hingegen Biergärten 7 und 8, so ergibt sich ein Saldo von $9,70 - 9,30 = 0,40$. Der Besuch der Biergärten 8 bis 10 ergäbe den Saldo 0, weil Sie nur Ausgaben hätten und diese Tour daher nicht in Frage kommt. Ebenso können Sie **nicht** eine Tour durch die Biergärten 1, 3 und 6 machen, da Sie nur nebeneinander liegende Biergärten besuchen dürfen.

Aufgabe — Teil a (7 Punkte)

Alles klar? Prima, dann ist die Aufgabenstellung schnell klar:

> Schreiben Sie ein Java-Programm, das die Beträge einer wie oben gezeigten Tabelle einliest und daraus den maximal möglichen Saldo errechnet. Ihr Programm muß dabei nach dem *Divide et Impera*- Prinzip vorgehen.

Beispiel: Die korrekte Ausgabe Ihres Programms bei Verwendung der obigen Tabelle muß 18,70 lauten.

Aufgabe — Teil b (3 Punkte)

> Erweitern Sie Ihre Lösung aus Teil a dahingehend, daß sie Ihnen auch den Anfangs- und Endbiergarten Ihrer Kneipentour angibt.

Beispiel: Bei Verwendung der obigen Daten ergibt sich der Saldo von 18,70, wenn man in Biergarten 3 beginnt und in Biergarten 7 aufhört.

Hinweise:

- Ihr Algorithmus muß in diesem Blatt unbedingt nach dem Prinzip *Divide et Impera* ablaufen.
- In den Übungen bekommen Sie noch einige Anregungen dazu, wie dieses Prinzip genau funktioniert.

Viel Erfolg — Und Prost!