Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr
Johannes Krugel, Jeremias Weihmann

Wintersemester 2010/11
Aufgabenblatt 10
12.01.2011

Praktikum Algorithmen-Entwurf

Letzter Abgabetermin: Dienstag, den 18.01.2011, 14:00 Uhr

Aufgabe 1 (Implementierung von Breakpoint-Graphen)

Implementieren Sie ein Programm, das folgenden Anforderungen genügt:

1. Der Input ist eine Datei, die in der ersten Zeile eine Permutation aus $\bar{\mathbf{S}}_n$ der Gestalt „$(\pi_1, \ldots, \pi_n)$“ mit $\pi_n = n$ enthält. Jede folgende nichtleere Zeile besteht aus zwei Paaren „$(i, j), (k, l)$“ mit $i, j, k, l \in [n] \cup (-[n])$. Im Folgenden bezeichnet π die zum aktuell betrachteten Breakpoint-Graphen korrespondierende Permutation mit $\pi_n = n$.

2. Das Programm stellt den Breakpoint-Graphen $G(\pi)$ visuell dar:

   (a) Die Knoten werden entsprechend der Definition gelabelt.

   (b) Die Knoten sind so angeordnet, dass der Reality-Obverse-Kreis einen Kreis im geometrischen Sinne approximiert.

   (c) Die beiden Knoten $-n(G)$ und $n(G)$ befinden sich an der unteren Seite des Kreises, wobei sich $-n(G)$ links neben $n(G)$ befindet.

   (d) Reality-Edges werden schwarz, Desire-Edges blau und Obverse-Edges rot dargestellt. Gibt es zwei Knoten x, y, sodass x und y sowohl durch eine Reality- als auch durch eine Desire-Edge verbunden sind, so werden diese beiden Kanten als eine gemeinsame grüne Kante dargestellt.

3. Das Programm bietet eine Methode an, die als Input zwei unterschiedliche Reality-Edges entgegennimmt. Die Methode führt die zu diesen Reality-Edges korrespondierende Reversion auf dem Breakpoint-Graphen $G(\pi)$ durch und gibt die entsprechende Reversion $r_{i,j}$ mit $j < n$ für π in Form des Paars (i, j) zurück. Als Nebeneffekt wird $r_{i,j}$ auf π angewandt. Sowohl (i, j) als auch $r_{i,j}(\pi)$ werden über den Standardoutputstream ausgegeben. Die Laufzeit dieser Methode liegt in $O(n + m)$.

4. Das Programm interpretiert die Paare $(i, j), (k, l)$ des Inputs als Reality-Edges des Breakpoint-Graphen. Die Zeilen des Inputs beginnend mit der zweiten Zeile werden nacheinander abgearbeitet. Für jede Zeile wird die zuvor beschriebene Methode mit den in der Zeile definierten Kanten aufgerufen. Dazu müssen natürlich zunächst die korrespondierenden Kanten des Graphen gefunden werden. Die Laufzeit für jede Zeile des Inputs liegt in $O(n + m)$.