
Praktikum Algorithmen-Entwurf

Letzter Abgabetermin: Dienstag, den 01.02.2011, 14:00 Uhr

Aufgabe 1 (Sortieren mit orientierten Reversionen)

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 „ (π_1, \dots, π_n) “ enthält. Da für diese Permutation nicht zwangsläufig $\pi_n = n$ gilt, muss die erweiterte Permutation $\pi' := (\pi_1, \dots, \pi_n, n+1)$ betrachtet werden. Im Folgenden bezeichnet π' die zum aktuell betrachteten Breakpoint-Graphen korrespondierende Permutation mit $\pi'_{n+1} = n+1$ und π die Permutation (π'_1, \dots, π'_n) .
2. Das Programm stellt den Breakpoint-Graphen $G := G(\pi')$ 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 zeigt die Werte $n(G), c(G), h(G), f(G)$ und $d(G) = n(G) - c(G) + h(G) + f(G)$ an.
4. Die Suche nach einer sicheren Reversion wird *nicht* visualisiert. Dies soll im Hintergrund stattfinden. Dazu kann beispielsweise eine Kopie des Graphen angelegt werden, auf der dann gearbeitet wird. Die Gesamtlaufzeit, um eine sichere Reversion zu finden, soll in $O(n(G)^3)$ liegen.
5. Wenn eine sichere Reversion gefunden wurde, wird diese auf den Breakpoint-Graphen angewandt. Weiterhin wird das Indexpaar (i, j) der zu dieser Reversion auf dem Breakpoint-Graphen korrespondierenden Reversion $r_{i,j}$ sowie die Permutation $r_{i,j}(\pi)$ ausgegeben. Beachten Sie, dass Sie dazu die beim letzten Mal implementierte Methode geringfügig modifizieren müssen, damit nicht $r_{i,j}(\pi')$, sondern $r_{i,j}(\pi)$ ausgegeben wird.