

## Übungsblatt 10

Abgabe bis Montag, den 17. Januar um 16 Uhr

### Aufgabe 1 (3 Punkte)

Fügen Sie der Klasse *Graph* aus der Vorlesung eine Methode *breadthFirstSearch* hinzu. Die Methode sollte zwei Argumente haben: den Startknoten und das Feld der Markierungen. Die Methode sollte ausgehend von diesem Startknoten eine Breitensuche zu allen erreichbaren nicht markierten Knoten ausführen und diese Knoten dabei markieren. Schreiben Sie auch einen Test für diese Methode.

### Aufgabe 2 (2 Punkte)

Fügen Sie der Klasse *Graph* eine Methode *computeConnectedComponents* hinzu, die mit Hilfe von *breadthFirstSearch* die Zusammenhangskomponenten des Graphen berechnet und die Größe der größten Zusammenhangskomponente zurückgibt. Schreiben Sie auch für diese Methode einen Test.

### Aufgabe 3 (2 Punkte)

Fügen Sie der Klasse *Graph* eine Methode *insertRandomEdges* hinzu, die einen zufälligen ungerichteten Graphen erzeugt. Die Methode sollte zwei Argumente haben: die Anzahl  $n$  der Knoten, und die Anzahl  $m$  der ungerichteten Kanten, wobei  $0 \leq m \leq n \cdot (n - 1)/2$ . Die Menge der ungerichteten Kanten des erzeugten Graphen sollten dann eine zufällige  $m$ -elementige Teilmenge der Menge aller  $n \cdot (n - 1)/2$  möglichen ungerichteten Kanten sein. Um eine ungerichtete Kante  $\{u, v\}$  einzufügen, fügen Sie einfach die beiden gerichteten Kanten  $(u, v)$  und  $(v, u)$  ein.

Schreiben Sie einen Test, der nachprüft (für ein festes  $n$ ), dass für  $m = 0$  keine Kanten eingefügt werden, und dass für  $m = n \cdot (n - 1)/2$  in der Tat alle möglichen Kanten eingefügt werden.

### Aufgabe 4 (2 Punkte)

Ein klassische Ergebnis aus der Graphentheorie besagt, dass ein zufälliger Graph, mit  $n$  und  $m$  wie oben beschrieben, für  $m > n/2$  mit sehr hoher Wahrscheinlichkeit eine sogenannte *giant component* hat, das heißt, eine Zusammenhangskomponente mit mindestens einem konstanten Bruchteil aller  $n$  Knoten. Für  $m < n/2$  dagegen enthält die größte Zusammenhangskomponente mit sehr hoher Wahrscheinlichkeit nur  $O(\log n)$  Knoten, also viel weniger als  $n$  Knoten.

Schreiben Sie ein Programm *GiantComponentMain*, welches diese Behauptung empirisch nachprüft. Verwenden Sie dabei die Methoden aus den Aufgaben vorher. Denken Sie sich selber geeignete Werte für  $n$  und  $m$  aus.

[bitte wenden]

**Aufgabe 5** (1 Punkt)

Committen Sie wie gehabt alles in das SVN (in einen neuen Unterordner *uebungsblatt-10*) und vergessen Sie nicht die *erfahrungen.txt* (in eben diesem Unterordner).