



Effiziente Algorithmen – Übung 2

Aufgabe 1 (JAVA-Graphen-Library)

Wir werden für die vielfältigen Graphenprobleme eine Klasse zur Repräsentierung ungerichteter Graphen benötigen. Wir brauchen folgende Funktionalitäten:

```
assign(...) // Zuweisung (G=G')
assign_complement(...) // Zuweisung des Komplements von G'
get_n() // Anzahl Knoten
get_m() // Anzahl Kanten
get_weight(...) // Gewicht einer Kante
set_weight(...) // Setzen des Gewichts einer Kante
get_degree(int i) // Knotengrad von Knoten i
get_degree() // Grad des Graphen = maximaler Knotengrad
assign_random_weights(...) // Konstruktion eines zufaelligen gewichteten
// Graphen mit geg. Kantenwahrscheinlichkeit
assign_random_edges(...) // Konstruktion eines zufaelligen ungewichteten Graphen
print_weights(...) // Drucken der Kantengewichte des Graphen
```

Entwickeln Sie eine Graphenklasse mit diesen Funktionen, wobei Sie die C++-Implementierung unter

http://www-crypto.htw-saarland.de/weber/teaching/03_ss_ea/

als Vorlage benutzen können.

Aufgabe 2 (Vertex Cover)

Implementieren Sie den in der Vorlesung besprochenen Algorithmus (Strategie der größten Knotengrade) für das vertex cover Problem inklusive Postprocessing-Schritt.