Technische Universität Wien
Institut für Computergraphik und Algorithmen
Arbeitsbereich für Algorithmen und Komplexität

186.866 Algorithmen und Datenstrukturen VU

Programmieraufgabe E8

---

1 Übersicht

Ihre Aufgabe ist es einen **2-Approximationsalgorithmus für Vertex Cover** zu implementieren. Das benötigte Framework ("AlgoDat_E8.zip") steht im TUWEL zur Verfügung. Implementieren Sie die Methode

- `Set<Integer> approxVertexCover(ADGraph G)`

in der Datei "src/exercise/E8.java".

Das Framework erwartet, dass `approxVertexCover` ein Set von Knoten zurück gibt, welches eine Approxiamtion des minimalen Vertex Covers des ungerichteten Graphen G darstellt. Ihr Algorithmus soll eine Approximationsgüte von 2 haben.

2 Implementierung

Die Datei "E8.java" enthält die Methodensignatur für ein approximiertes minimales Vertex Cover. Die Rückgabe ist ein Set mit Vertex Ids. Ein solches Set können Sie zum Beispiel mit

- `Set<Integer> var = new HashSet<>()`

anlegen. Diese Datenstruktur unterstützt Einfügen und Suchen in $O(1)$. Weitere Informationen zum "Set"-Interface finden Sie in der Dokumentation der Java-Standardbibliothek

- `https://docs.oracle.com/javase/8/docs/api/java/util/Set.html`.

Der Graph wird als Objekt der Klasse "ADGraph" repräsentiert. Auf die Knoten/Kanten können Sie über die Knoten/Kanten-Ids zugreifen, welche als `int` gespeichert sind. Die Ids der Knoten/Kanten laufen von 0 (inklusive) bis `numVertices()` (exklusive) bzw. `numEdges()` (exklusive).

Die Klasse "ADGraph" beinhaltet die folgenden, für Sie relevanten Methoden. Dies bedeutet **nicht**, dass Sie alle diese Funktionen verwenden müssen, um die Aufgabe erfolgreich zu lösen!

- Die folgenden Methoden haben konstante Laufzeit:
  - `int degree(int v)`
    Gibt den Grad von v zurück.
  - `int source(int e)`
    Sei e eine Kante mit Knoten (u, v), diese Methode gibt u zurück.
  - `int target(int e)`
    Sei e eine Kante mit Knoten (u, v), diese Methode gibt v zurück.
  - `int numVertices()`
    Gibt die Anzahl Knoten im Graphen zurück.
  - `int numEdges()`
    Gibt die Anzahl Kanten im Graphen zurück.
- Diese Methoden haben schlimmstenfalls lineare Laufzeit in der Größe des Graphen:
  - `ArrayList<Integer> neighbors(int v)`
    Gibt die Nachbarn von v zurück.
  - `ArrayList<Integer> incidentEdges(int v)`
    Gibt alle zu v inzidenten Kanten zurück.
- Diese Methoden sind **ausschließlich zum Debuggen** gedacht:
  - `Set<Integer> getNonCoveredEdges(Set<Integer> vertices)`
    Gibt das Set von Kanten zurück, die nicht von den Knoten in `vertices` überdeckt werden.
  - `Set<String> edgesToString(Set<Integer> edges)`
    Gibt für jede Kante $e = (u, v)$ in `edges` eine String Repräsentation der Form "(u, v)" zurück.

3 Auswertung

Wenn Sie die Datei "AlgoDat_E8.java" ausführen, können Sie, wie im Tutorial beschrieben, in der Konsole eine Testinstanz auswählen. Es wird eine *.csv-Datei* im *solutions* Ordner erstellt, welche Sie mit der Datei "E8/Evaluation/plot.html" auswerten können.

Hier wird ein Plot generiert, der folgendes plottet:

$$\frac{\Sigma(|c_a| - |c_o|)}{|I_s|}$$

wobei c_a das approximierte Vertex-Cover ist, c_o das optimale Vertex-Cover, und I_s das Set der Instanzen der Größe s.

Weiter wird pro Graph eine SVG Datei erstellt, die den Graphen und das von Ihnen berechnete Vertex Cover visualisiert. Diese Dateien werden auch geschrieben, wenn eine falsche Lösung berechnet wird. In diesem Fall sind im SVG natürlich entsprechend eine inkorrekte Auswahl an Knoten markiert.

Die Graphen aus der Datei "rome_ad_29.csv" sind ein Standardbenchmark mit bis zu 29 Knoten. Die Datei "rome_ad_33.csv" beinhaltet Graphen mit bis zu 33 Knoten. Die Graphen aus der Datei "test.csv" sind drei relativ kleine, zufällige Graphen, sowie ein Graph mit genau einer Kante und ein Dreieck.

Fügen Sie für die Abgabe den Plot für "rome_ad_33.csv" ein und beantworten Sie folgende Fragen:

- Wie haben Sie die 2-Approximation umgesetzt? Beschreiben Sie ihre Implementierung.
- Beschreiben Sie den Plot. Sind die geplotteten Werte immer kleiner gleich 2? Wenn Nein, warum nicht?
- Was müssten Sie plotten, um tatsächlich die Approximationsgüte ihrer Implementierung zu untersuchen? Geben Sie die Formel und eine Begründung an.

Hinweis: Bevor Sie beginnen, lesen Sie sich unbedingt das Tutorial zu den Programmieraufgaben durch („Tutorial.pdf"). Dieses finden Sie im TUWEL. Sie bietet eine kurze Einführung in das verwendete Framework.