Abschlussprojekt: Wiki-Graph

Einführung in Python

emanuel.barth@uni-jena.de

Jedes Abschlussprojekt soll selbstständig bearbeitet und gegebenenfalls auch präsentiert werden können. Module die nicht zur Standardinstallation von Python gehören, dürfen nur dann verwendet werden, wenn die Aufgabenstellung diese explizit erlaubt. Abgegeben wird der ausführlichst kommentierte und ausführbare Pythoncode. Sollte es Fragen zur Aufgabenstellung geben könnt ihr mir einfach eine Mail schreiben.

Aufgabenstellung:

Im Jahr 2001 wurde das Wikipedia-Projekt offiziel gestartet und hat es sich zum Ziel gesetzt, die größte und umfangreichste, mehrsprachige Enzyklopädie der Welt zu werden. Mit ca. 61 Millionen verschiedenen Artikeln (Stand 2023), hat sie ihr Ziel längst erreicht und gehört zu den Top 10 der meist besuchtesten Webseiten weltweit. Innerhalb eines Wikipedia-Artikels sind meist Links zu weiterführenden Artikeln enthalten. Diese Verknüpfungen lassen sich als gerichteten Graphen darstellen, wobei jeder Knoten einen Artikel repräsentiert und eine gerichtete Kante von einem Knoten **X** zu einem zweiten Knoten **Y** gezogen wird, wenn im Artikel von **X** eine Verlinkung nach **Y** vorhanden ist.

Implementiert ein lauffähiges Python Kommandozeilenskript, welches folgendes leisten soll:

- Eingabe ist eine URL zu einem Wikipedia-Artikel, welche als Startpunkt des Wiki-Graphen dienen soll.
- Mittels des Moduls *request* (oder einem Ähnlichen Modul) kann der HTML-Quellcode des Wikipedia-Artikels runtergeladen und analysiert werden.
  Hier beginnt nun der iterative Aufbau des Wiki-Graphs. Die aktuelle Seite soll in den Graphen eingetragen werden und für alle auf ihr verlinkten Artikelseiten sollen nun ebenfalls der Quellcode geladen und entsprechend analysiert werden. Der Vorgang wird erst dann abgebrochen, wenn mindestens eine der folgenden Bedingungen eintritt:
  - Eine maximale Anzahl **K** an Knoten im Graph ist erreicht.
  - Die maximale Distanz **D** (hier: Verlinkungsschritte) zum Ursprungsartikel darf nicht überschritten werden.

- Der Graph soll selbständig objektorientiert implementiert werden, d. h. es sollen eigene Klassen mit dazugehörigen Attributen und Methoden geschrieben werden. Dafür dürfen auch die bereits in der Vorlesung erstellten Graphklassen verwendet werden.
- Die Ausgabe des Graphs kann dann in Form einer Matrix erfolgen, welche in geeigneter Form in eine Datei gespeichert werden soll.
- Stellt den Wiki-Graph auch visuell als Bild dar. Dafür dürfen beliebige schon vorhandene Pythonmodule zur Visuallisierung von Graphen benutzt werden.
- Die Parameter **K** und **D** sollen als Argumente anpassbar und per Default mit $K = 500$ und $D = 10$ belegt sein.
- Einfache Graphoperationen sollen implementiert sein, mindestens aber: Das Abfragen aller Nachbarknoten zu einem gegebenem Knoten, die Rückgabe der Knoten mit dem höchsten Eingangs- und Ausgangsgrad und die Ausgabe der Dichte des Graphs.
- Es soll möglich sein, den Graphen nach gerichteten und ungerichteten Kreisen zu durchsuchen. Ein Kreis ist eine Folge von Knoten, die alle unterschiedlich sind, wobei der erste und letzte Knoten identisch sind. Alle gefundenen Kreise sollen in geeignert Textform und visuell ausgegeben werden.
- Zu jedem Knoten (welche ja einem Wikipedia-Artikel entsprechen) sollen die 10 wichtigsten Schlüsselwörter des Artikels extrahiert und in einer Liste gespeichert werden. Schlüsselwörter sind hierbei Substantive, die im Artikel besonders häufig vorkommen und somit eine hohe Relevanz haben. Die Schlüsselwörter sollen auch in geeigneter Form gespeichert werden.

Hinweis:

Nicht alle Verlinkungen im Quellcode eines Wikipedia-Artikels führen zu einem neuen Wikipedia-Artikel, diese sollen ignoriert werden.

Bonus:

- Nutze Bibliotheken wie `Plotly` oder `Bokeh`, um interaktive Graphen zu erstellen, die es ermöglichen, Knoten zu klicken und weitere Informationen anzuzeigen, wie z.B. die Distanz zum Eingabeknoten, sein Grad, ob der Knoten Teil eines Kreises ist, seine Schlüsselwörter, etc.
- Nutzt Techniken der Parallelisierung mit Hilfe des `multiprocessing` Moduls, um den Graphenaufbau zu beschleunigen.