27. Bundeswettbewerb
Informatik 2008/2009
Endrunde 22.-25.9.2009

Suchmaschine

Gruppenarbeit 2. Tag

Ziel dieser Aufgabe ist, eine Suchmaschine fürs Internet zu bauen (ohne dabei auf andere Suchmaschinen wie z.B. Google zurückzugreifen). Dazu müssen folgende Aufgaben bewältigt werden:

1. **Durchforsten des Internets (Stichwort: *Web-Graph*)**

   Ein *Graph G* ist eine mathematische Struktur, bestehend aus einer Menge von *Knoten V* und einer Menge von *Kanten E*. Eine Kante ist ein Knotenpaar (A, B), so dass man sich eine Kante als Verbindung von Knoten A zu Knoten B vorstellen kann.

   Die "Link-Struktur" des Internets kann man sich als einen solchen Graphen vorstellen: Jede Webseite (d.h. jede URL) ist ein Knoten, und es gibt eine Kante von Webseite A zu Webseite B, falls die Webseite A einen Link zur Webseite B enthält. Dieser Graph wird *Web-Graph* genannt.

   Überlegen Sie sich, wie man auf systematische Art Informationen über den Web-Graphen erhalten und in einer geeigneten Datenstruktur abspeichern kann.

2. **Indexierung der gefundenen Seiten**

   Entwickeln Sie eine Datenstruktur (den so genannten *invertierten Index*), mit deren Hilfe man für jedes beliebige Wort t eine Liste erhält, die angibt, welche Webseite das Wort t mit welcher Häufigkeit enthält. Geben Sie einen möglichst effizienten Algorithmus an, der eine solche Datenstruktur anlegt.

3. **Abschätzen des "Renommees" einer Webseite (Stichwort: *PageRank*)**

   Wir wollen jeder Webseite A eine Zahl $PR_A \geq 0$ namens *PageRank von A* zuweisen, die angibt, wie "wertvoll" (im Sinne von "wichtig", "seriös", bzw. "einflussreich") diese Webseite ist. Die Zahl PR_A soll umso größer sein, je höher das Renommee der Webseite A ist. Das Renommee und damit der PageRank PR_B einer Webseite B wird als hoch eingeschätzt, wenn viele Webseiten A mit hohem PageRank PR_A einen Link auf Seite B enthalten. Es hat sich als vorteilhaft erwiesen, die Werte PR_A, die allen Webseiten A zugeordnet werden, folgendermaßen zu wählen:

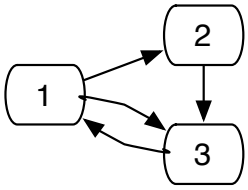
   - Eine Webseite A mit ausgehenden Links auf k_A weitere Seiten "vererbt" ihren PageRank anteilig an jede der k_A Webseiten B, auf die sie mit einem Link verweist (jeweils um den Anteil $\frac{PR_A}{k_A}$).

- Aus praktischen Gründen wird die "Vererbung" des PageRanks einer Seite A auf Seiten, auf die sie mit einem Link verweist, um den Dämpfungsfaktor $\frac{1}{2}$ abgeschwächt. Insgesamt setzt sich der PageRank PR_B einer Webseite B folgendermaßen zusammen:

$$PR_B = \frac{1}{2 \cdot N} + \frac{1}{2} \cdot V_B,$$

  wobei N die Gesamtzahl aller Webseiten ist und V_B die Summe der Werte $\frac{PR_A}{k_A}$ über alle Webseiten A ist, die einen Link auf Seite B enthalten.

Ermitteln Sie zunächst den PageRank für das kleine Beispiel, in dem der Web-Graph aus nur drei Webseiten 1, 2 und 3 besteht, wobei Webseite 1 Links auf Seiten 2 und 3 enthält, Webseite 2 nur einen Link auf Seite 3 enthält, und Webseite 3 nur einen Link auf Seite 1 enthält.



Entwickeln Sie dann einen Algorithmus, der bei Eingabe eines beliebigen Web-Graphen $G = (V, E)$ den PageRank aller Webseiten bestimmt.

4. Beantwortung von Suchanfragen

Bei Eingabe eines oder mehrerer Such-Stichworte soll eine Liste aller Seiten ausgegeben werden, die relevante Informationen zu den eingegebenen Stichworten enthalten. Diese Liste soll so sortiert sein, dass die "besten Treffer" am weitesten oben in der Liste stehen.

Entwickeln Sie einen Algorithmus zur Beantwortung solcher Suchanfragen.

Gehen Sie dabei insbesondere darauf ein, auf welche Art Sie bewerten, wie "relevant" eine Webseite für ein Such-Stichwort ist. Dabei ist es sinnvoll, jeder Webseite A und jedem Such-Stichwort t eine Zahl namens $Score_{A,t}$ zuzuordnen, die die Relevanz von Webseite A für das Suchwort t angibt. Der Wert $Score_{A,t}$ sollte dabei auf geeignete Art aus den in den Punkten (1) bis (3) generierten Informationen berechnet werden können. Entwickeln Sie verschiedene Ansätze zur Wahl von $Score_{A,t}$ und diskutieren Sie jeweils deren Vor- und Nachteile.

Versuchen Sie, Ihren Algorithmus so zu erweitern, dass auch Suchanfragen beantwortet werden können, die aus *Booleschen Kombinationen* von Suchworten bestehen (z.B. die Anfrage "Finde alle relevanten Seiten, die die Worte t_1, t_2 und (t_3 oder t_4), aber keins der Worte t_5 und t_6 enthalten.").

Beachten Sie, dass der Web-Graph riesengroß ist und dass Suchanfragen innerhalb weniger Sekunden beantwortet werden müssen. Dazu stehen Ihnen große Cluster aus vielen herkömmlichen PCs zur Verfügung, die Sie für jede der oben genannten Aufgaben (1)–(4) auf geeignete Weise nutzen sollten. Geben Sie jeweils auch Abschätzungen über die von Ihren Verfahren benötigte Laufzeit und den Speicherplatzbedarf an.