Eerste deeltoets DAR
27 Mei 2016
Uitwerkingen

Opgave 2: Top-k

(i) Je kunt TA niet gebruiken, want dan heb je een index op alle tabellen nodig. Je kunt NRA wel gebruiken, maar dat is niet volledig efficiënt. Het beste is een hybride aanpak: van de oid's die je in tabel B tegenkomt zoek je de bijbehorende waarde in tabel A op. Andersom doe je dat niet, omdat je geen index op tabel B hebt.

(ii) De eerste 2 ronden van het hybride algoritme:

Round	1	2
Max-A	200	160
Max-B	200	160
Threshold	400	320
Buffer	[2:320] [5:200-400]	*[5:360] *[2:320] [3:160-320]
Top-k		[5:360] [2:320]

(iii) F is nu monotoon dalend in B. Sorteer kolom B van laag naar hoog.

Opgave 3: Google Pagerank

$$G = \alpha H + \alpha \frac{1}{n} e a^\top + (1-\alpha)\frac{1}{n} e e^\top$$

H is een sparse matrix, dus voor de vermenigvuldiging van H met P kunnen we sparse matrix technieken gebruiken. De matrix $ee^\top$ is niet sparse, maar laten we die eens met P vermenigvuldigen:

$$ee^\top P = \begin{pmatrix} \sum_{i=1}^n P_i \\ \vdots \\ \sum_{i=1}^n P_i \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix},$$

omdat $\sum_{i=1}^{n} P_i = 1$. Belangrijke constatering is dat alle elementen van de resulterende vector hetzelfde zijn, dus je hoeft maar één keer $(1-\alpha)/n$ uit te rekenen (is natuurlijk voor alle iteraties hetzelfde).

Hetzelfde geldt voor:

$$ea^{\top}P = \begin{pmatrix} \sum P_d \\ \vdots \\ \sum P_d \end{pmatrix},$$

waarbij we sommeren over de "dangling nodes". Wederom zijn alle elementen van de resulterende vector hetzelfde, dus we hoeven er maar één uit te rekenen. Dit moet wel iedere iteratie opnieuw gebeuren.