FACULTE INFORMATIQUE ET COMMUNICATIONS
SCHOOL OF COMPUTER AND COMMUNICATION SCIENCES

Laboratoire de Systèmes d'Information Repartis
Distributed Information Systems Laboratory

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Distributed Information Systems

Prof. Karl Aberer

Final exam, Spring semester 2013 / 2014
June 27th 2014, 12:15 – 15:15

The following materials are allowed: lecture slides, exercise sheets and solutions, past exams with your own solution, and personally written notes. You can use a pocket calculator but any other electronic devices (including mobile phones, laptops, handheld devices, etc.) **must be switched off**. The exam consists of 29 pages including the cover sheet. Please write your answers *only* on the appropriate pages.

- If necessary, you can ask for additional sheets.
- Do not separate the exam sheets (by unstapling).
- Please number the additional pages and do not forget to put your name on them.

Student name: **Karl Aberer** Seat number: **0**

PLEASE HAVE YOUR STUDENT CARD READY FOR CONTROL

GOOD LUCK!

Each question receives a maximum of **25 points**.

Question 1	Question 2	Question 3	Question 4	Total points

Question 1: Mobile Data Broadcasting

Question 1.1 (15 pts)

SwissTXT is providing the teletext for the swiss TV channels. They provide information in several domains such as "News", "Sports" or "Travels". The table below shows the page numbers per category, assuming all pages are used from the first to the last page for each category. The gaps between categories are empty.

Category	First page	Last page
News	100	149
Sports	200	274
TV/Radio	300	399
Travel	400	429
Finance	500	549
Entertainment	600	629

From a recent survey, they concluded that 45% of the consumers are interested in the News, 30% in the Sports, 10% in the TV/Radio and 5% in each of Travel, Finance and Entertainment. If we assume that all customers would access the pages in their favorite category with the same frequency and uniform randomly, what should be the optimal broadcast schedule for SwissTXT?

a) Give the list of broadcast disks and their optimal frequencies.

b) Give the list of chunks and their contents (to simplify, express them as ranges of pages, e.g. 20-29).

c) Give the optimal schedule in terms of chunks.

Question 1.2 (10 pts)

Let's suppose we want to add a single index at the beginning of a flat broadcast disk (without replication) of 256 pages, knowing that an index page is the same size as a data page. We test 3 versions: one with a binary tree, another one with a tree with fanout=4 and one with a fanout=16, and we assume a uniform distribution of data accesses.

1. Which one of the following propositions is true? [*Justify briefly your answer!*]
   - ☐ *a*) Their latencies are all equal.
   - ☐ *b*) The latency of the one with fanout=16 is the largest.
   - ☐ *c*) The latency of the one with fanout=4 is the largest.
   - ☐ *d*) The latency of the one with fanout=2 is the largest.
2. Which one of the following propositions is true? [*Justify briefly your answer!*]
   - ☐ *a*) Their tuning times are all equal.
   - ☐ *b*) The tuning time of the one with fanout=16 is the smallest.
   - ☐ *c*) The tuning time of the one with fanout=4 is the smallest.
   - ☐ *d*) The tuning time of the one with fanout=2 is the smallest.

Question 2: Graph Databases

Given the data graph D and schema graph S_1, represented in Figure 1 below, and whose adjacency lists appear in Table 1 and Table 2 respectively.

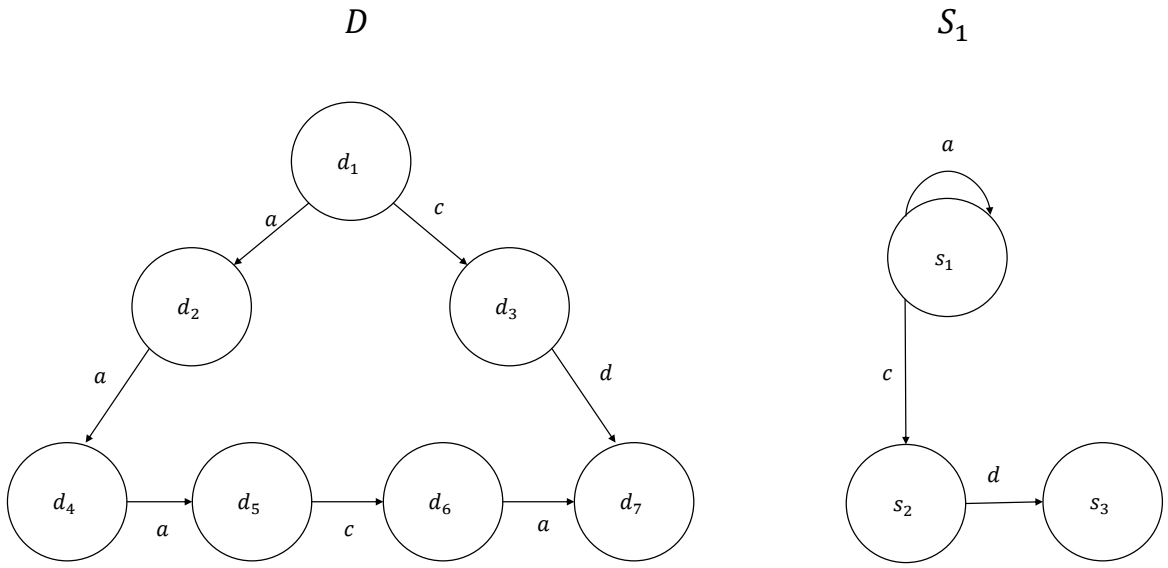

Figure 1: Data Graph D and schema graph S_1

Node 1	Node 2	Edge Label
d_1	d_2	a
d_2	d_4	a
d_4	d_5	a
d_5	d_6	c
d_6	d_7	a
d_1	d_3	c
d_3	d_7	d

Table 1: Adjacency list for data graph D

Node 1	Node 2	Edge Label
s_1	s_1	a
s_1	s_2	c
s_2	s_3	d

Table 2: Adjacency list for schema graph S_1

Question 2.1 (3 pts)

Relationship R_1 defined as $(D <_{R_1} S_1)$ is not satisfied because: [*Justify briefly your answer!*]

☐ a) S_1 cannot simulate the path $a \to a \to c \to a$.

☐ b) D cannot simulate the path $a \to a \to a \to a$.

☐ c) S_1 cannot simulate the path $a \to a \to a \to c \to a$.

☐ d) D cannot simulate the path $a \to a \to a \to c \to d$.

Question 2.2 (3 pts) Keeping all the current nodes and edges in S_1 intact, give a new schema graph S_2 such that:

- S_2 adds to S_1 exactly one directed edge (or one alternative label to an existing edge)
- S_2 simulates D.

Question 2.3 (4 pts) Give a classification of the nodes of D implied by the relationship R_2 ($D <_{R_2} S_2$). (It doesn't need necessarily to be the one corresponding to the maximal simulation.)

Question 2.4 (5 pts) Give the data guide DG of D.

Question 2.5 (5 pts) Give a schema S_3 with a minimum number of nodes that simulates D.

Question 2.6 (5 pts) Which one of the following is true? (NB: R_3 is defined in the opposite direction compared to the usual case.) [*Justify briefly your answer!*]

- ☐ *a*) ($S_3 <_{R_3} D$) is satisfied.
- ☐ *b*) ($S_3 <_{R_3} D$) is not satisfied because S_3 does not simulate the path $a \to a \to a \to c \to a$
- ☐ *c*) ($S_3 <_{R_3} D$) is not satisfied because D does not simulate the path $a \to c \to a$
- ☐ *d*) None of the above

Question 3: Information Retrieval

There are four sorted posting lists for four terms "football", "Brazil", "Germany" and "Belgium" shown in the table. Each item in the posting list is of the form $\{$*document id, TF-IDF weight*$\}$. These four posting lists are distributed among four nodes in the network. There is another node hosting no posting lists, which is referred to as requesting node. The items of the posting lists are sent to the requesting node for query processing.

Different from the conventional data access method of **Fagin's algorithm** which retrieves the items of one posting list one by one, a block-based data access method fetches m items at a time of one posting list. For instance, given $m = 2$, if the scan phase would stop at the third item of one posting list, the last block of items of that posting list transferred to the requesting node would be the third and fourth item of the posting list, but the requesting node only processes the third item. In the random access phase, the requested item is still transferred one by one. But, the items that have been transferred to the requesting node via the block-based data access in the scan phase don't need to be retrieved again through random access.

Football	Brazil	Germany	Belgium
$\{1,0.85\}$	$\{4,0.96\}$	$\{8,0.94\}$	$\{3,0.95\}$
$\{3,0.8\}$	$\{3,0.96\}$	$\{10,0.91\}$	$\{5,0.89\}$
$\{4,0.78\}$	$\{5,0.91\}$	$\{12,0.90\}$	$\{7,0.88\}$
$\{8,0.73\}$	$\{7,0.85\}$	$\{4,0.85\}$	$\{8,0.83\}$
$\{7,0.71\}$	$\{6,0.79\}$	$\{3,0.78\}$	$\{1,0.81\}$
$\{6,0.65\}$	$\{1,0.78\}$	$\{5,0.75\}$	$\{4,0.75\}$
$\{11,0.63\}$	$\{12,0.72\}$	$\{6,0.72\}$	$\{10,0.73\}$
$\{10,0.6\}$	$\{2,0.63\}$	$\{1,0.65\}$	$\{6,0.66\}$
$\{9,0.5\}$	$\{8,0.6\}$	$\{7,0.61\}$	$\{11,0.54\}$
$\{12,0.49\}$	$\{9,0.56\}$	$\{2,0.58\}$	$\{12,0.43\}$
$\{2,0.45\}$	$\{11,0.4\}$	$\{9,0.43\}$	$\{9,0.4\}$
$\{5,0.3\}$	$\{10,0.31\}$	$\{11,0.37\}$	$\{2,0.32\}$

Question 3.1 (10 pts)

Compute the top-k documents using Fagin's algorithm with the block-based data access method for query q = "football Brazil Germany" where $k = 3$ and $m = 2$. Assume an aggregation function that returns the maximum of the tf-idf scores. Analyse the number of data items transferred for each posting list to the requesting node.

Question 3.2 (15 pts)

In this question, we assume that the requesting node has a cache to store the retrieved items of the posting lists for one query. Then, when another query is processed, the requesting node can first access this local cache to fetch required items, if they are available in the cache (otherwise it will retrieve them through the network).

Now the requesting node caches the retrieved items of the query in Question 3.1. Assume that the cache is used in the processing of the following query: q_1 = "football Brazil Germany Belgium", issued by the requesting node with $k = 2$. Determine the items transferred through network to the requesting node for processing q_1 with the assistance of the cache. The block-based data access method is still applied in the scan phase with m=2.

Question 4: Recommender Systems

In this question, we want to build a distributed recommender system for Web credibility assessment.

We propose to combine 3 approaches in order to establish the rating of unknown pages. The approaches are based on:

- Existing ratings of pages by different users
- Different features based on the content of the pages
- The links existing between these pages

For each of the following subquestions, every step of the computation and the detailed thought process that led to such steps are requested.

	$User1$	$User2$	$User3$	$User4$
Page A	5	2	4	4
Page B		2	3	4
Page C	5	3	1	3
Page D	2		4	4
Page E	4	1		5

Table 3: Rating of the 5 pages by 4 users in a five-level Likert scale (from 1 to 5)

User	*neighbors*
User 1	User 2, User 3
User 2	User 1, User 4
User 3	User 1
User 4	User 2

Table 4: Neighborhood of each user

Page	*Sentiment*	*Topic*	*Keywords*
A	2	sport	{2014, Brazil, football, CocaCola, sponsor, worldcup}
B	1	sport	{2014, ball, Brazil, football, results, worldcup}
C	4	sport	{ball, Federer, Rolex, sponsor, tennis }
D	3	business	{ingredients, drink, Pepsi, sugar_free}
E	3	business	{Brazil, CocaCola, drink, Pepsi, sponsor, worldcup}

Table 5: Content feature of each page

Question 4.1 (8 pts)

The users of the distributed recommender system are organized in a social network to share their ratings. For the sake of computation efficiency and in order to limit the number of messages sent through the network, we define the neighborhood of each user as the set of users she is directly connected to in this network. All the neighborhoods are given in Table 4.

Using the ratings available in Table 3 and based on the neighborhoods defined in Table 4, determine the missing ratings (in Table 3) using a **user-based** collaborative filtering with the Pearson similarity measure. *Hint: you could start by building a variation of Table 3, taking into account the average rating of the users.*

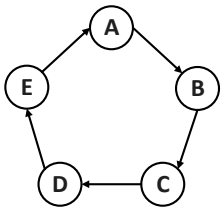

Figure 2: Links between pages.

Hub	*Linked to authority*
A	B
B	C
C	D
D	E
E	A

Table 6: Adjacency list equivalent to Figure 2

Question 4.2 (8 pts)

The second component is evaluating the content of the page: perform a **content-based** recommendation for the missing ratings in Table 3 using the known ratings of each users. You will use the following parameters:

- Pages belong to the same neighborhood if they share the same *topic* according to Table 5.
- The similarity measure between two pages is computed as follows:

$$sim(p1,p2) = sentiments(p1,p2) * Jaccard(p1,p2)$$

Where $sentiments(p1,p2)$ assesses the closeness of documents p1 and p2 in term of sentiment. It is defined as follow:

$$sentiments(p1,p2) = 1 - \frac{|p1.sentiment - p2.sentiment|}{4}$$

and where $Jaccard(p1,p2)$ corresponding to the Jaccard similarity between the sets of keywords of p1 and p2 (Jaccard similarity is computed as the size of the intersection of two sets divided by the size of their union).

$$Jaccard(p1,p2) = \frac{|p1.keywords \bigcap p2.keywords|}{|p1.keywords \bigcup p2.keywords|}$$

Question 4.3 (8 pts)

The last component is based on the HITS algorithm. The pages at hand have the relationship defined in Figure 2 (or its adjacency list equivalence in Table 6). In a first step, compute the normalized *Hub* and *Authority* scores of each page. In a second step, generate a rating for the pages B, D, E taken as a weighted combination of the hub and authority scores of these pages with the respective weights: 2 (for the hub score) and 3 (for the Authority score):

$$LinkBasedRating(p) = 1 + HubScore + 3 * AuthorityScore$$

The weighting is motivated by the assumption in our system that an Authority should be more credible than a Hub. NB: This rating is independent of the user.

Question 4.4 (1 pts)

Finally, compute for each missing rating the average of the three different components, obtained respectively in question 4.1, 4.2 and 4.3. This rating will be the final output of our recommender system.