Distributed Information Systems: Spring Semester 2016

Quiz 6

Student Name: _____ Date: 26 May 2016

Student ID: _____ Time: 11:15AM to 11:30AM

Total number of questions: 8

Each question has a single answer!

1. What does the following map-reduce program compute:

```
Mapper:
        def mapper(document, line):
                foreach word in line.split():   # split(): splits the line into words
                        output(1, len(word))  # len(string): returns the length of a string.

Combiner:
        def combiner(key, values):output(key, sum(values))

Reducer:
        def reducer(key, values): output(key, sum(values))
```

☐ *a*) Computes the total number of words in all documents

☐ *b*) Computes the total number of documents

☒ *c*) Computes the total number of characters in all words in all documents

☐ *d*) None of the above

2. Using paragraph-level granularity as compared to document-level granularity during index construction results in:

☒ *a*) Less post-processing

☐ *b*) Smaller index

☐ *c*) Both A and B

☐ *d*) None of the above

3. Which of the following is an **advantage** of the tf-idf ranking scheme?

☐ *a*) It accounts for semantically similar words (e.g., synonyms).

☐ *b*) It accounts for the position of a word in the document.

☐ *c*) It ignores small syntactic differences among words.

☒ *d*) It reduces the relative importance of very frequent words.

4. Which of the following is **true** in the context of inverted files:

☐ *a*) The number of index terms is generally much larger than the number of occurrences of these terms in the documents.

☐ *b*) Index merging compresses an inverted file index on disk and reduces storage cost.

☒ *c*) The trie structure used for index construction can also be used as a data access structure to terms in the vocabulary.

☐ *d*) The finer the addressing granularity used in documents, the smaller the posting file becomes.

5. Which of following is **false** about Vector Space Retrieval?

☐ *a*) It represents both documents and queries in the same vector space.

☐ *b*) It can be based on different types of term weighting schemes.

☐ *c*) Given a query, it can produce ranked results in terms of similarity.

☒ *d*) Documents can never be retrieved if they don't contain all query keywords.

6. Which of following is **false** about Fagin's algorithm?

☐ *a*) The elements in the posting lists are sorted.

☐ *b*) The order in which the posting lists are processed does not influence the final result.

☒ *c*) The result is independent of the weight aggregation function used. can use max or sum

☐ *d*) The algorithm might terminate without performing any random accesses to the posting lists.

7. Using the cosine similarity based on tf-idf, as used in vector space retrieval, which of the following documents has/have the **highest similarity** to the query q="badger"?

d1	d2	d3	d4
The badger is a cousin of the wolverine.	badger badger badger badger badger badger	badger	Wolverine smokes the cigar.

☐ *a*) d1

☒ *b*) d2, d3

☐ *c*) d2

☐ *d*) d1, d3

8. We want to return, from the two posting lists below, the top-2 documents matching a query using Fagin's algorithm with the aggregation function taken as the sum of the tf-idf weights. How many entries (total of both lists) are accessed in the **first phase** of the algorithm (i.e., before performing the random access)?

EPFL	
document	tf-idf
d3	0.8
d2	0.6
d1	0.5
d4	0.4

Lausanne	
document	tf-idf
d4	0.8
d1	0.6
d3	0.5
d2	0.4

☐ *a*) 2

☐ *b*) 4

☒ *c*) 6

☐ *d*) 8