

---

**Midterm (30 pts)**

## 1 Positional Index and Querying (2 + 6)

Consider the following fragment of a positional index with the format:

word: doc#: <posn, posn, ...>; doc#: <posn, ...>.

Gates: 1: <3>; 2: <6>; 3: <2,17>; 4: <1>.

IBM: 4: <3>; 7: <14>.

Microsoft: 1: <1>; 2: <1,21>; 3: <3>; 5: <16,22,51>.

The  $/k$  operator, `word1 /k word2` finds occurrences of `word1` within `k` words of `word2` (on either side), where `k` is a positive integer argument. Thus `k = 1` demands that `word1` be adjacent to `word2`.

- Describe the set of documents that satisfy the query: `Gates /2 Microsoft`.
- Describe each set of values for `k` for which the query: `Gates /k Microsoft` returns a different set of documents as the answer.

## 2 True/False with Justification (2\*3 + 2\*2)

In Information Retrieval Systems:

1. Stemming never lowers precision.
2. Case folding decreases the size of the dictionary.
3. Normally, stop words elimination reduces the maximum length of the (positional) postings list.
4. To properly account for co-references (e.g., pronouns, acronyms, aliases, etc.), tf-idf scores should be increased appropriately in their presence.

*Precision* is defined as the number of *relevant* documents retrieved by a search divided by the *total* number of documents retrieved by that search. *Recall* is defined as the number of *relevant* documents retrieved by a search divided by the total number of *existing relevant* documents (which should have been retrieved).

### 3 Estimating Time for Sorting (4 + 2)

If we need  $T \log_2 T$  comparisons (where  $T$  is the number of termID-docID pairs) and two disk seeks for each comparison, how much time would it take to sort 1 million terms if we used disk for storage and an unoptimized sorting algorithm (i.e., not an external sorting algorithm)? What is the closest time unit that captures the order of magnitude of the time required: a microsecond, a second, an hour, a day, a month or a year?

Repeat this calculation for the case when the complete data is in the main memory and using in-memory sort?

Use the system parameters reproduced below. ( $\log_2 10 = 3.3$  and  $\log_{10} 2 = 0.3$ )

Symbol	Statistic	Value
m	memory transfer time per byte	5 ns
s	average seek time	5 ms
p	typical ALU operation	10 ns

### 4 Citation Count Task (6)

Given a directed graph as an adjacency list: `Ids -> list(Ids)`

```
s1 : s11, s12, ...
s2 : s21, s22, ...
...
```

determine the list of pairs  $(sn, refs)$ , such that `refs` is the number of links incident on `sn`, using MapReduce paradigm. E.g., for the following graph, the expected pairs are  $(s1, 0)$ ,  $(s2, 1)$ ,  $(s3, 2)$ ,  $(s4, 3)$ .

```
s1 : s2, s4.
s2 : s3, s4.
s4 : s3, s4.
```

Explain clearly the *map* task and the *reduce* task, defining these functions using *set*-notation.