INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
Mid Semester Examination 2024-25

Sub No: CS60092 **Sub Name:** Information Retrieval
Department/Centre/School : **Computer Science and Engineering**
Duration: 2 hrs **Marks: TBD**

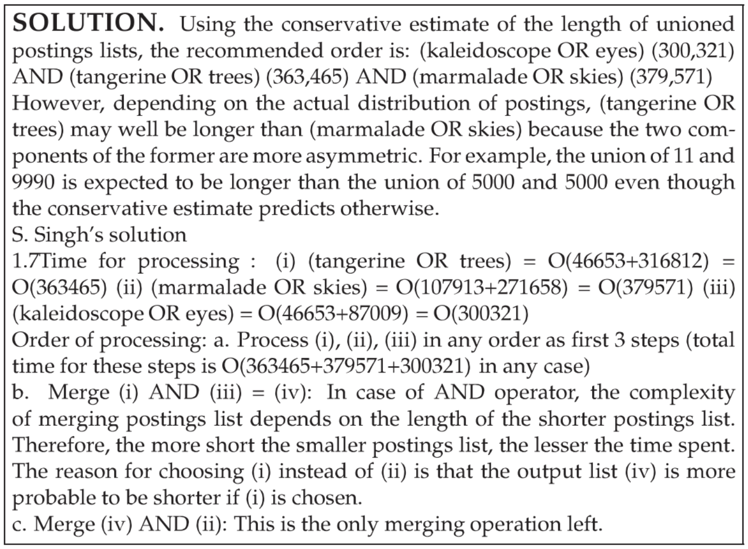
Specific charts, graph paper, log book, etc. required: NO

Special Instructions: ANSWER as much as you can. **All parts of a single question should be answered together.** Answers should be brief and to-the-point. All calculations must be shown. No marks will be awarded for sketchy answers and answers/results without proper reasoning/calculations. In case of reasonable doubt, make assumptions and state them upfront. You can keep probability values in fractional forms.

1. Recommend a query processing order for the query
`(tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)`
if the sizes of the postings lists of these words are as follows.

Terms	**Postings size**
eyes	213312
kaleidoscope	87009
skies	107913
marmalade	271658
trees	46653
tangerine	316812

Justify your answers by giving estimates of the time required for each sub-segment of the query processed. **[4]**



SOLUTION. Using the conservative estimate of the length of unioned postings lists, the recommended order is: (kaleidoscope OR eyes) (300,321) AND (tangerine OR trees) (363,465) AND (marmalade OR skies) (379,571) However, depending on the actual distribution of postings, (tangerine OR trees) may well be longer than (marmalade OR skies) because the two components of the former are more asymmetric. For example, the union of 11 and 9990 is expected to be longer than the union of 5000 and 5000 even though the conservative estimate predicts otherwise.

S. Singh's solution

1.7Time for processing : (i) (tangerine OR trees) = O(46653+316812) = O(363465) (ii) (marmalade OR skies) = O(107913+271658) = O(379571) (iii) (kaleidoscope OR eyes) = O(46653+87009) = O(300321)

Order of processing: a. Process (i), (ii), (iii) in any order as first 3 steps (total time for these steps is O(363465+379571+300321) in any case)

b. Merge (i) AND (iii) = (iv): In case of AND operator, the complexity of merging postings list depends on the length of the shorter postings list. Therefore, the more short the smaller postings list, the lesser the time spent. The reason for choosing (i) instead of (ii) is that the output list (iv) is more probable to be shorter if (i) is chosen.

c. Merge (iv) AND (ii): This is the only merging operation left.

2. State true or false with justification.

(a) In Boolean retrieval system stemming never lowers precision.

(b) In Boolean retrieval system stemming never lowers recall.

(c) Stemming increases the size of the vocabulary.

(d) Stemming should be invoked only at index construction time.

$[\mathbf{1.5 \times 4 = 6}]$

SOLUTION. a. False. Stemming can increase the retrieved set without increasing the number of relevant docuemnts. b. True. Stemming can only increase the retrieved set, which means increased or unchanged recall. c. False. Stemming decreases the size of the vocabulary. d. False. The same processing should be applied to documents and queries to ensure matching terms.

3. Let us consider a two word query. One of the words has the following postings list with 16 entries as follows
`[4, 6, 10, 12, 14, 16, 18, 20, 22, 32, 47, 81, 120, 122, 157, 180]`
while the other word has only one entry
`[47]`
Work out how many comparisons are needed to intersect the two postings lists using the following two strategies. Justify your answers in each case.

   (a) Using standard postings list.

   (b) Using postings list with skip pointers with skip length $\sqrt{L}$ where L is the length of the postings list.

$[\mathbf{2.5 + 2.5 = 5}]$

SOLUTION.
Applying MERGE on the standard postings list, comparisons will be made unless either of the postings list end i.e. till we reach 47 in the upper postings list, after which the lower list ends and no more processing needs to be done.
Number of comparisons = 11
b. Using skip pointers of length 4 for the longer list and of length 1 for the shorter list, the following comparisons will be made: 1. 4 & 47 2. 14 & 47 3. 22 & 47 4. 120 & 47 5. 81 & 47 6. 47 & 47 Number of comparisons = 6

4. Consider the following two documents:
Doc 1: `new home sales top forecast`
Doc 2: `home sales rise in july`

   (a) Build positional indexes on top of these documents using the format `DocID: <(position1, position2), ... >`. For example, the positional index for the word "sales" is as follows: `sales: 1:<3>; 2: <2>`
   new: 1:<1>; 2: <1> home: 1:<2>; 2: <1> . . .

   (b) Return all the docs and corresponding positions for which the query conditions are met. If none of the documents meet the criteria, return `none`.

      i. `new home` – **1:** $<1,2>$

      ii. `new sales` – **none**

      iii. `new /2 sales` **1:** $<1,3>$

[2 + 3 = 5]

5. Answer the following two questions.

   (a) Let the relevance labels of the first 10 documents for a query be–
   `1, 0, 0, 1, 1, 0, 1, 0, 0, 0.`
   Here, 1 means relevant, and 0 means non-relevant. Suppose the total number of relevant documents for this query is 5.

   i. Draw the Precision-Recall curve for this result set.
   ii. Write down the general formula for finding the interpolated precision value at recall level 'r'.
   iii. Draw the interpolated precision curve.
   iv. What is the average precision for this result set?

   (b) Suppose a user has three different types of information needs and enters queries accordingly. Which one among the three evaluation measures– *Precision*, *Recall* and *Reciprocal Rank* should be used to measure the effectiveness of the result sets for each of the following query types?

   i. Website of IIT Kharagpur.
   ii. Searching for free high quality wallpapers.
   iii. Information about Cardiovascular Disease.
   iv. Looking at research done in a particular area before preparing a research proposal.

[(3+3+2+1.5) + (4 x 1) = 13.5]

6. Suppose that a document collection consists of following two documents.
Doc 1: `free eBooks free software eBooks`
Doc 2: `hundred free pdfs`
The user's initial query is –
Q: `free eBooks free pdfs free computer eBooks`
The user judges **Doc 1** relevant and **Doc 2** non-relevant. Assume that we are using direct term frequency (with no scaling and no document frequency). There is no need to length-normalize vectors. Using Rocchio relevance feedback, what would the revised query vector be after relevance feedback? Assume $\alpha = 1$, $\beta = 0.75$, $\gamma = 0.25$.

[6]

7. Consider the query– **Q:** `catholic church brisbane`. Assume the vector space model and the TF-IDF weighting scheme. The corpus consists only of the following documents:
Doc 1: `roman catholic church brisbane roman`
Doc 2: `church brisbane church church`
Doc 3: `catholic catholic protestant protestant all all brisbane`
Doc 4: `catholic church welcome catholic church brisbane`

   Assign vector indices 1 – 7 to the vocabulary words in the following order:
   `roman, catholic, church, brisbane, protestant, all, welcome`

   (a) Now, assuming this order, clearly write the complete weight vectors for the query and each document. State the formula used for IDF.

   (b) What are the lower and upper bounds for IDF of a term in a corpus according to the stated formula?

(c) Rank the documents w.r.t. the query assuming the overlap score measure [Hint: The score for a query-document pair is the sum of the weights of the query terms in the document vector].

(d) Rank the documents assuming the cosine similarity as a scoring function. Show all steps of the computation.

[(3 + 1 + 2 + 4) = 10]

8. Given the following relevance judgements R_{J1}, R_{J2} in Table 1, calculate κ measure between two judges:

		R_{J2}		Total
		Yes	No	
R_{J1}	Yes	300	20	320
	No	10	70	80
Total		310	90	400

Table 1: Relevance Judgements

[5]

9. In the Binary Independence Model (BIM), we rank documents according to their Retrieval Status Value (RSV$_d$), where RSV is defined as:

$$RSV_d = \sum_{t:x_t=q_t=1} \log\left(\frac{p_t(1-u_t)}{u_t(1-p_t)}\right)$$

Suppose we have the following query–

`q: dangling pointer`

Given the following information about each of the query terms in Table 2:

Term	N(x$_t$=1, R=1) = s (Number of relevant documents in which term is present)	N(R=1) = S (Number of relevant documents)	df$_t$ (Document frequency of the term)
`dangling`	5	20	10
`pointer`	20	50	40

Table 2: Information about Query terms

Total number of documents in the collection, N = 100. For logarithm, use base 10.
Calculate RSV of the following document.

`d: dangling pointer is dangerous`

[5]