Birla Institute of Technology & Science, Pilani
Work-Integrated Learning Programs Division
Second Semester 2018-2019

Mid-Semester Test
(EC-2 Make-Up Solutions)

Course No.	: SS ZG537
Course Title	: INFORMATION RETRIEVAL
Nature of Exam	: Closed Book
Weightage	: 30%
Duration	: 2 Hours
Date of Exam	: / / 2019

No. of Pages = 8
No. of Questions = 7

Note:
1. Please follow all the *Instructions to Candidates* given on the cover page of the answer book.
2. All parts of a question should be answered consecutively. Each answer should start from a fresh page.
3. Assumptions made if any, should be stated clearly at the beginning of your answer.

Q1. [3 + 5 =8 marks]

a) Discuss in brief the limitations of the Jaccard coefficient.
b) Discuss briefly the index construction algorithm used in logarithmic merge in Dynamic Indexing with a suitable diagram. What is the time complexity of logarithmic merge?

Ans.

a)

- It doesn't consider *term frequency* (how many times a term occurs in a document)
- Rare terms in a collection are more informative than frequent terms. Jaccard doesn't consider this information (capital of India) (GDP of India) (capital india)
- We need a more sophisticated way of normalizing for length since Jaccard coefficient does not consider length normalized document vectors.

(1 mark for each point)

b)

- Maintain a series of indexes, each twice as large as the previous one.
- Keep smallest (Z_0) in memory
- Larger ones (I_0, I_1, …) on disk
- If Z_0 gets too big ($> n$), write to disk as I_0
- or merge with I_0 (if I_0 already exists) as Z_1
- Either write merge Z_1 to disk as I_1 (if no I_1)
- Or merge with I_1 to form Z_2 etc. ---1 mark

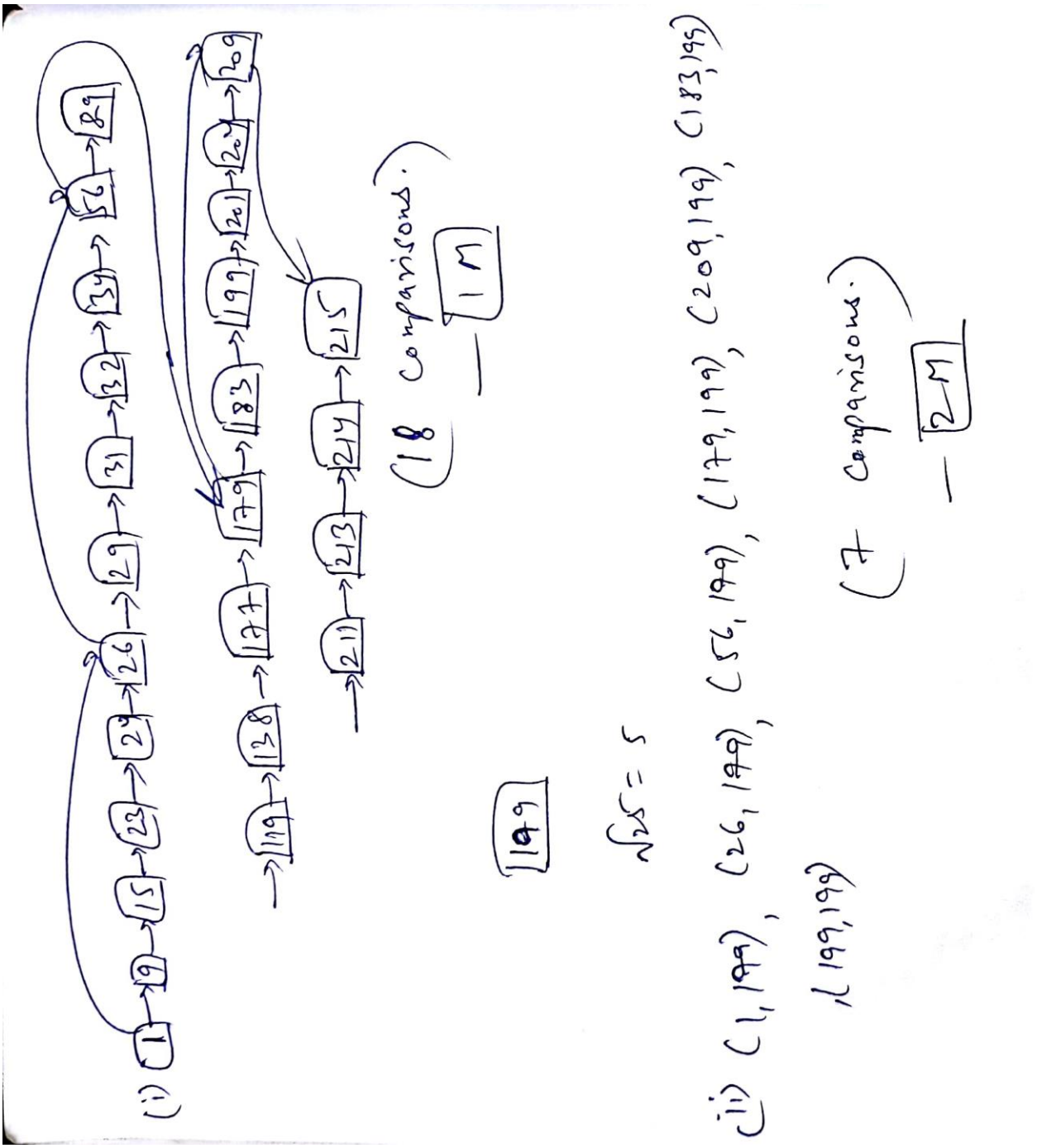
```
LMERGEADDTOKEN(indexes, Z_0, token)
 1  Z_0 ← MERGE(Z_0, {token})
 2  if |Z_0| = n
 3    then for i ← 0 to ∞
 4         do if I_i ∈ indexes
 5               then Z_{i+1} ← MERGE(I_i, Z_i)
 6                    (Z_{i+1} is a temporary index on disk.)
 7                    indexes ← indexes − {I_i}
 8               else I_i ← Z_i    (Z_i becomes the permanent index I_i.)
 9                    indexes ← indexes ∪ {I_i}
10                    BREAK
11         Z_0 ← ∅

LOGARITHMICMERGE()
1  Z_0 ← ∅    (Z_0 is the in-memory index.)
2  indexes ← ∅
3  while true
4  do LMERGEADDTOKEN(indexes, Z_0, GETNEXTTOKEN())
```

--3 marks

Each posting is merged O(log T) times, so complexity is O(T log T) --1 mark

Q2. Given a two-word query. The postings list of one term consists of the following 25 entries: [1, 9, 15, 23, 24, 26, 29, 31, 32, 34, 56, 89, 119, 138, 177, 179, 183, 199, 201, 204, 209, 211, 213, 214, 215] and for the other it is the one entry postings list: [199].
How many comparisons would be done to intersect the two postings lists using skip pointers, with a skip length as discussed in the class. [3 marks]

Ans.

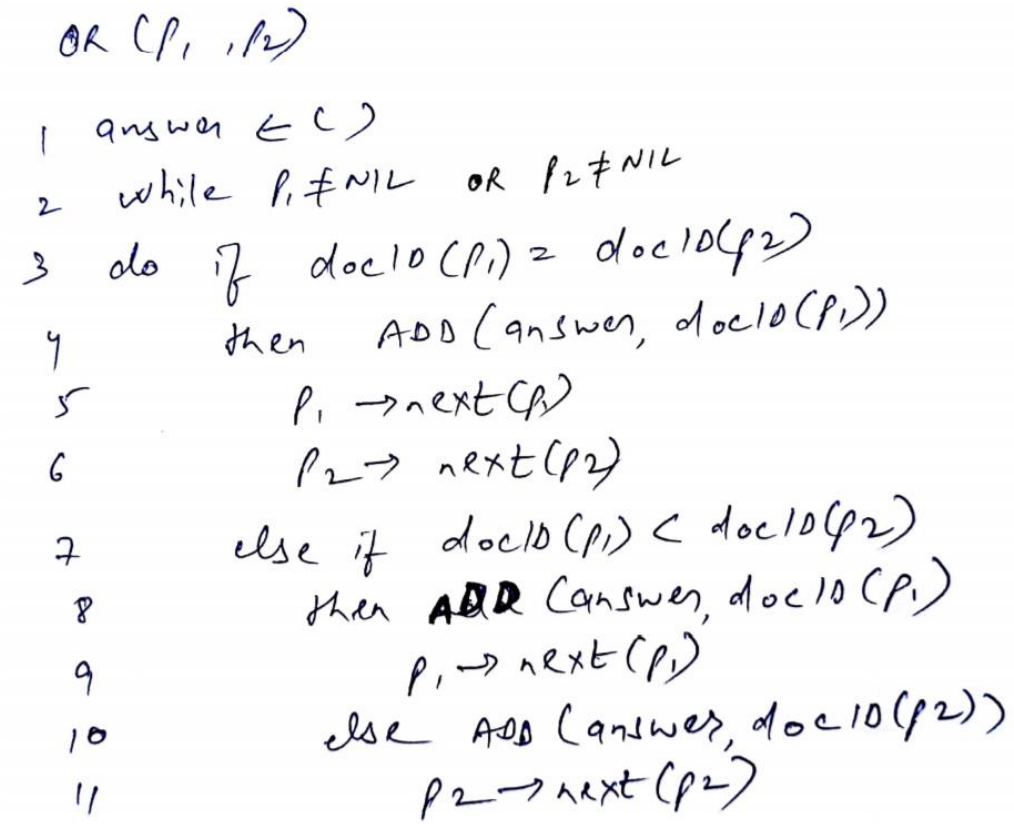


Q3. [2 + 2 = 4 marks]

a) Write the pseudocode for merging the postings list of “Brutus” and “Caesar” in the query “Brutus OR Caesar”.

b) Suppose a program for recognizing dogs in scenes from a video identifies 9 dogs in a scene containing 11 dogs and some cats. If 4 of the identifications are correct, but 5 are actually cats, then compute the precision and recall of the program.

Ans. a)



(2 marks)

b) precision = 4/9, recall = 4/11 (1 mark each)

Q4. [2+1=3 marks]

a) Consider a collection made of 500000 documents, each containing on average 800 words. The number of different terms is estimated to 700000 and the average length of a non-positional posting list is 200. If 20 fixed bytes are used for terms, 4 bytes for term frequency and 4 bytes for storing the pointer to postings list, compute the memory usage for dictionary and Postings list.

b) What is the soundex code for the following two names, Michael and Michele? Assume that the alphabets are mapped to numbers as follows: *(B, F, P, V → 1), (C, G, J, K, Q, S, X, Z → 2), (D,T → 3), (L → 4), (M, N → 5)* and *(R → 6)* . Show the computations.

Ans: a) Postings list size= Total bytes /Term = 20+4+4 = 28 bytes ---1 mark
Dictionary size= 700000 * 28 bytes = 19.6 Mega bytes ---1 mark

b) Michael - M240 ; Michele – M240; (0.5 mark each)

Q5. Compute the minimum edit distance using Levenshtein algorithm, between the terms WHAT and WASTE. Fill in the table given below by distances between all prefixes as computed by the algorithm. [2 marks]

		W	H	A	T
	0	1	2	3	4
W	1				
A	2				
S	3				
T	4				
E	5				

Ans.

		W	H	A	T
	0	1	2	3	4
W	1	0	1	2	3
A	2	1	1	1	2
S	3	2	2	2	2
T	4	3	3	3	2
E	5	4	4	4	3

Edit distance =3
(0.5 mark for each column)

Q6. Consider that you are given a task to filter incoming mails as spam or non-spam. You have a database of a set of mails with their class (i.e. spam or non-spam) where a set of words are used as feature to classify a mail to be one of these types. Let say the words be A, B, C and D; and the class is represented as **S** or **NS**. [3+3 =6 marks]

A	B	C	D	Type
3	1	0	2	NS
2	0	1	1	NS
1	1	1	1	NS
4	1	1	0	NS
0	1	0	0	NS
0	2	5	0	S
1	3	4	4	S
2	0	4	5	S
1	0	0	8	S
4	1	0	7	S

a) For the given problem above, generate a Naïve-Bayes classification model by assuming the occurrence of the words (A, B, C and D) as Bernoulli's trial. Do not apply add-one smoothing in the formulation of the conditional probabilities.

b) Using the classifier which you just modeled in question (a), classify the following e-mails as spam or non-spam with the following features: E-mail2(0,2,6,0).

Ans.

a) Bernoulli's trial as occurrence of the words means the following: if a word occurs (i.e. if number of occurrence is > 0), then it is considered 1; otherwise 0. (1 means occurs, 0 means does not occur). With this transformation, our new dataset looks like this:

A	B	C	D	Type
1	1	0	1	NS
1	0	1	1	NS
1	1	1	1	NS
1	1	1	0	NS
0	1	0	0	NS
0	1	1	0	S
1	1	1	1	S
1	0	1	1	S
1	0	0	1	S
1	1	0	1	S

For Naïve-Bayes, we need compute the following probabilities:

P(NS)=5/10

P(S)=5/10 --- 1 mark

P(A=1|NS)=4/5 P(A=0|NS)=1-P(A=1|NS)=1-4/5 (Optional)

P(B=1|NS)=4/5

P(C=1|NS)=3/5

P(D=1|NS)=3/5 --- 1 mark

Similarly,

P(A=1|S)=4/5 P(A=0|S)=1-P(A=1|S)=1-4/5 (Optional)

P(B=1|S)=3/5

P(C=1|S)=3/5

P(D=1|S)=4/5 ---- 1 mark

[If the student has computed these probabilities: 1 mark for each group as indicated (only for all correct answers within that group; otherwise 0 for wrong value).]

b) Naïve-Bayes testing for new mails:

E-mail2(0,2,6,0)=E-mail2(0,1,1,0)

P(mail=NS|E-mail2)

= P(mail=NS|A=0,B=1,C=1,D=0)

= P(NS)×P(A=0|NS)×P(B=1|NS)×P(C=1|NS)×P(D=0|NS)

= P(NS)×[1-P(A=1|NS)]×P(B=1|NS)×P(C=1|NS)×[1-P(D=1|NS)]

= 0.0192 --- 1 mark

P(mail=S|Email2)

= P(mail=S|A=0,B=1,C=1,D=0)

= P(S)×P(A=0|S)×P(B=1|S)×P(C=1|S)×P(D=0|S)

= P(S)×[1-P(A=1|S)]×P(B=1|S)×P(C=1|S)×[1-P(D=1|S)]

= 0.0072 --- 1 mark

The probability for NS is higher than that of S. So, the E-mail2 will be classified as NS. ---1 mark

[1 mark for each class probability computation. 1 mark for last reasoning and decision.]

Q7. Given below are two tables, Ist table gives the **tf** values and IInd table gives the **idf** values for the 4 terms and 2 documents. Compute the Cosine similarity between Doc1 and Doc2. Use logarithmic tf, idf and cosine normalization for the computation. (**ltc**) [4 marks]

Term	Doc1	Doc2
T1	15	5
T2	2	22
T3	0	22
T4	3	0

idf values

Term	df*t*	idf*t*
T1	2312	0.64
T2	345	1.46
T3	3030	0.52
T4	178	1.75

Ans:

	Doc1					Doc2				
Terms	tf-raw	tf-wt	**idf**	**tf.idf**	**n'lize**	tf-raw	tf-wt	**idf**	**tf.idf**	**n'lize**
T1	15	2.18	0.64	**1.4**	**0.4**	5	1.7	0.64	**1.09**	**0.29**
T2	2	1.3	1.46	**1.9**	**0.54**	22	2.34	1.46	**3.42**	**0.9**
T3	0	0	0.52	**0**	**0**	22	2.34	0.52	**1.22**	**0.32**
T4	3	1.48	1.75	**2.59**	**0.74**	0	0	1.75	**0**	**0**

Cos(Doc1, Doc2) = 0.4 * 0.29 + 0.54*0.9 = 0.6

The table computation for each doc: 2 marks
Similarity computation: 2 marks

****