University of Texas at Dallas Department of Computer Science CS6322 – Information Retrieval Spring 2022

Instructor: Dr. Sanda Harabagiu

Mid-Term Exam

Issued: March 10th 2022 at Noon Due: March 12th 2022 – before Midnight

**Instructions**: Do not communicate with anyone in any shape or form. This is an independent exam. Do not delete any problem formulation, just write your answer immediately after each question in the space provided. **You are required to type your answers**. If the problem is deleted and you send only the answer, you shall receive ZERO points. If you do not write your name and netid, it will be considered that you did not submit your midterm exam, and will obtain ZERO points.

Copy and paste the Midterm Exam into a Word document, enter your answers (either by typing in MS Word) and transform the file of the exam into a PDF format. Submit the PDF file with the name **MidTerm\_Exam\_netID.pdf**, where netID is your unique netid provided by UTD. If you submit your exam in any other format your will receive ZERO points. If you handwrite your answers you shall receive ZERO points.

The MidTerm exam shall be submitted in eLearning before the deadline. No late submissions shall be graded! Any cheating attempt will determine the ENTIRE grade of the final exam to become ZERO.

**Submit in eLEarning as PDF file**

**DO NOT DELETE ANY PROBLEM, Simply add your answers!!!!!**

If you submit only the solutions with no problems, you will receive 0 points!!!!

**Problem 1 : (Stemming)** (**2 points**)

# *Compute the measure required by the Porter Stemmer for the following strings:*

1. enormously (**1 point**)
2. year (**1 point**)

Hint: show all details of how you obtained the measure!

*SOLUTION* 1.(a):

E N O R M O U S L Y

| | | | | | | | | |

V C V C C V V C C C

| | | | | |

V C V C V C

[VC] [VC] [VC]

[C] [VC]m [C]

m = 3

1.(b): Y E A R

| | | |

C V V C

| | |

C V C

[C] [VC]m [V]

m=1

**Problem 2 : (Tolerant Retrieval)** (**10 points**)

# *You have created the index of a collection of documents resulting in:*

allow  2  4  6 figuratively  5 6 friendly  1  3  5

show  1  2  4  6

shower  2  5

showroom  3  4  5

# *You would like to allow tolerant retrieval for searching this collection. What permuterms should you consider?*

*Detail your solution. (****6 points****)*

# *Hint: Show which permuterms were obtained for each term.*

1. *What documents are retrieved when processing the following queries:*

Q1: \*ow OR \*ly Q2: sho\* AND \*er Q3: f\*ly

Q4:\*ra\*

# *Provide ALL the details that allowed you to find the retrieved documents (****4 points****)*

*SOLUTION* 2.(a):

The permuterms of *allow* are:

allow$

llow$a

low$al

ow$all

w$allo

$allow

The permuterms of *figuratively* are:

figuratively$

iguratively$f

guratively$fi

uratively$fig

ratively$figu

atively$figur

tively$figura

ively$figurat

vely$figurati

ely$figurativ

ly$figurative

y$figurativel

$figuratively

The permuterms of *friendly* are:

friendly$

riendly$f

iendly$fr

endly$fri

ndly$frie

dly$frien

ly$friend

y$friendl

$friendly

The permuterms of *show* are:

show$

how$s

ow$sh

w$sho

$show

The permuterms of *shower* are:

shower$

hower$s

ower$sh

wer$sho

er$show

r$showe

$shower

The permuterms of *showroom* are:

showroom$

howroom$s

owroom$sh

wroom$sho

room$show

oom$showr

om$showro

m$showroo

$showroom

*SOLUTION* 2.(b):

1. \*ow -> search for permuterm ow$\*

Terms: ow$all : 2,4,6 OR ow$sh : 1,2,4,6

Result: 1,2,4,6

OR

\*ly -> search for permuterms ly$\*

Terms: ly$figurative : 5,6 OR ly$friend : 1,3,5

Result: 1,3,5,6

Answer: 1,2,3,4,5,6

1. sho\* -> search for permuterm $sho\*

Terms: $show : 1,2,4,6 OR $shower : 2,5 OR $showroom: 3,4,5

Result: 1,2,3,4,5,6

AND

\*er -> search for permuterms er$\*

Terms: er$show : 2,5

Result: 2,5

Answer: 2,5

1. f\*ly -> search for permuterms ly$f\*

Terms: ly$figurative : 5,6 OR ly$friend : 1,3,5

Answer: 1,3,5,6

1. \*ra\* -> search for permuterms ra\*

Terms: ratively$figu : 5,6

Answer: 5,6

**Problem 3 : (Spelling Correction)** (**10 points**)

1. *Compute the edit distance between “commander” and “commotion”. Give all the details of the computation.*

(**9 points**)

1. *What is the SOUNDEX code for “*Spencer*”. Show who you have obtained the code! (1* ***point****)*

*SOLUTION* 3.(a):

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | C | O | M | M | A | N | D | E | R |
|  | |  |  | | --- | --- | |  |  | |  | 0 | | |  |  | | --- | --- | |  |  | | 1 | 1 | | |  |  | | --- | --- | |  |  | | 2 | 2 | | |  |  | | --- | --- | |  |  | | 3 | 3 | | |  |  | | --- | --- | |  |  | | 4 | 4 | | |  |  | | --- | --- | |  |  | | 5 | 5 | | |  |  | | --- | --- | |  |  | | 6 | 6 | | |  |  | | --- | --- | |  |  | | 7 | 7 | | |  |  | | --- | --- | |  |  | | 8 | 8 | | |  |  | | --- | --- | |  |  | | 9 | 9 | |
| C | |  |  | | --- | --- | |  | 1 | |  | 1 | | |  |  | | --- | --- | | 0 | 2 | | 2 | 0 | | |  |  | | --- | --- | | 2 | 3 | | 1 | 1 | | |  |  | | --- | --- | | 3 | 4 | | 2 | 2 | | |  |  | | --- | --- | | 4 | 5 | | 3 | 3 | | |  |  | | --- | --- | | 5 | 6 | | 4 | 4 | | |  |  | | --- | --- | | 6 | 7 | | 5 | 5 | | |  |  | | --- | --- | | 7 | 8 | | 6 | 6 | | |  |  | | --- | --- | | 8 | 9 | | 7 | 7 | | |  |  | | --- | --- | | 9 | 10 | | 8 | 8 | |
| O | |  |  | | --- | --- | |  | 2 | |  | 2 | | |  |  | | --- | --- | | 2 | 1 | | 3 | 1 | | |  |  | | --- | --- | | 0 | 2 | | 2 | 0 | | |  |  | | --- | --- | | 2 | 3 | | 1 | 1 | | |  |  | | --- | --- | | 3 | 4 | | 2 | 2 | | |  |  | | --- | --- | | 4 | 5 | | 3 | 3 | | |  |  | | --- | --- | | 5 | 6 | | 4 | 4 | | |  |  | | --- | --- | | 6 | 7 | | 5 | 5 | | |  |  | | --- | --- | | 7 | 8 | | 6 | 6 | | |  |  | | --- | --- | | 8 | 9 | | 7 | 7 | |
| M | |  |  | | --- | --- | |  | 3 | |  | 3 | | |  |  | | --- | --- | | 3 | 2 | | 4 | 2 | | |  |  | | --- | --- | | 2 | 1 | | 3 | 1 | | |  |  | | --- | --- | | 0 | 2 | | 2 | 0 | | |  |  | | --- | --- | | 1 | 3 | | 1 | 1 | | |  |  | | --- | --- | | 3 | 4 | | 2 | 2 | | |  |  | | --- | --- | | 4 | 5 | | 3 | 3 | | |  |  | | --- | --- | | 5 | 6 | | 4 | 4 | | |  |  | | --- | --- | | 6 | 7 | | 5 | 5 | | |  |  | | --- | --- | | 7 | 8 | | 6 | 6 | |
| M | |  |  | | --- | --- | |  | 4 | |  | 4 | | |  |  | | --- | --- | | 4 | 3 | | 5 | 3 | | |  |  | | --- | --- | | 3 | 2 | | 4 | 2 | | |  |  | | --- | --- | | 1 | 1 | | 3 | 1 | | |  |  | | --- | --- | | 0 | 2 | | 2 | 0 | | |  |  | | --- | --- | | 2 | 3 | | 1 | 1 | | |  |  | | --- | --- | | 3 | 4 | | 2 | 2 | | |  |  | | --- | --- | | 4 | 5 | | 3 | 3 | | |  |  | | --- | --- | | 5 | 6 | | 4 | 4 | | |  |  | | --- | --- | | 6 | 7 | | 5 | 5 | |
| O | |  |  | | --- | --- | |  | 5 | |  | 5 | | |  |  | | --- | --- | | 5 | 4 | | 6 | 4 | | |  |  | | --- | --- | | 3 | 3 | | 5 | 3 | | |  |  | | --- | --- | | 3 | 2 | | 4 | 2 | | |  |  | | --- | --- | | 2 | 1 | | 3 | 1 | | |  |  | | --- | --- | | 1 | 2 | | 2 | 1 | | |  |  | | --- | --- | | 2 | 3 | | 2 | 2 | | |  |  | | --- | --- | | 3 | 4 | | 3 | 3 | | |  |  | | --- | --- | | 4 | 5 | | 4 | 4 | | |  |  | | --- | --- | | 5 | 6 | | 5 | 5 | |
| T | |  |  | | --- | --- | |  | 6 | |  | 6 | | |  |  | | --- | --- | | 6 | 5 | | 7 | 5 | | |  |  | | --- | --- | | 5 | 4 | | 6 | 4 | | |  |  | | --- | --- | | 4 | 3 | | 5 | 3 | | |  |  | | --- | --- | | 3 | 2 | | 4 | 2 | | |  |  | | --- | --- | | 2 | 2 | | 3 | 2 | | |  |  | | --- | --- | | 2 | 3 | | 3 | 2 | | |  |  | | --- | --- | | 3 | 4 | | 3 | 3 | | |  |  | | --- | --- | | 4 | 5 | | 4 | 4 | | |  |  | | --- | --- | | 5 | 6 | | 5 | 5 | |
| I | |  |  | | --- | --- | |  | 7 | |  | 7 | | |  |  | | --- | --- | | 7 | 6 | | 8 | 6 | | |  |  | | --- | --- | | 6 | 5 | | 7 | 5 | | |  |  | | --- | --- | | 5 | 4 | | 6 | 4 | | |  |  | | --- | --- | | 4 | 3 | | 5 | 3 | | |  |  | | --- | --- | | 3 | 3 | | 4 | 3 | | |  |  | | --- | --- | | 3 | 3 | | 4 | 3 | | |  |  | | --- | --- | | 3 | 4 | | 4 | 3 | | |  |  | | --- | --- | | 4 | 5 | | 4 | 4 | | |  |  | | --- | --- | | 5 | 6 | | 5 | 5 | |
| O | |  |  | | --- | --- | |  | 8 | |  | 8 | | |  |  | | --- | --- | | 8 | 7 | | 9 | 7 | | |  |  | | --- | --- | | 6 | 6 | | 8 | 6 | | |  |  | | --- | --- | | 6 | 5 | | 7 | 5 | | |  |  | | --- | --- | | 5 | 4 | | 6 | 4 | | |  |  | | --- | --- | | 4 | 4 | | 5 | 4 | | |  |  | | --- | --- | | 4 | 4 | | 5 | 4 | | |  |  | | --- | --- | | 4 | 4 | | 5 | 4 | | |  |  | | --- | --- | | 4 | 5 | | 5 | 4 | | |  |  | | --- | --- | | 5 | 6 | | 5 | 5 | |
| N | |  |  | | --- | --- | |  | 9 | |  | 9 | | |  |  | | --- | --- | | 9 | 8 | | 10 | 8 | | |  |  | | --- | --- | | 8 | 7 | | 9 | 7 | | |  |  | | --- | --- | | 7 | 6 | | 8 | 6 | | |  |  | | --- | --- | | 6 | 5 | | 7 | 5 | | |  |  | | --- | --- | | 5 | 5 | | 6 | 5 | | |  |  | | --- | --- | | 4 | 5 | | 6 | 4 | | |  |  | | --- | --- | | 5 | 5 | | 5 | 5 | | |  |  | | --- | --- | | 5 | 5 | | 6 | 5 | | |  |  | | --- | --- | | 5 | 6 | | 6 | 5 | |

The edit distance is 5.

Operations: c o m m a n d e r

| | | | |

Replace Replace Replace Replace Replace

| | | | |

c o m m o t i o n

*SOLUTION* 2.(b):

Soundex for Spencer is S152.

From the Soundex – typical algorithm:

1. Retain the first letter of the word: S (Rule-1)
2. Change letters to digit: p -> 1 (Rule-3)
3. Change letters to digit: e -> 0 (Rule-2)
4. Change letters to digit: n -> 5 (Rule-3)
5. Change letters to digit: c -> 2 (Rule-3)
6. Change letters to digit: e -> 0 (Rule-2)
7. Change letters to digit: r -> 6 (Rule-3)
8. Remove all 0s: S105206 -> S1526 (Rule-5)
9. Return first four positions: S152 (Rule-6)

**Problem 4 :** (**54 points**)

# *Consider the following three short documents:*

Doc #1

Probiotics help digest the milk sugar lactose. Doc #2

Probiotics in yogurt may boost the functioning of the immune system. Doc #3

If you want to give your yogurt a nutritional upgrade, go for Greek yogurt.

* 1. (**18 points**) Tokenize manually the document collection, and identify the terms after you remove the following stop words: “the”, “a”, “in”, “of”, “may”, “to”, “for”, “your” . In this way “probiotics” becomes “probiotic”, “functioning” becomes “function” etc.
     1. Generate the dictionary of the collection and list it, adding also the document frequency (**5 points**)

Use the format

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Index | 1 | 2 | … |  |  |  |
| Term | boost | … |  |  |  |  |
| DF | 1 | … |  |  |  |  |

* + 1. Represent the 3 documents as document vectors by computing three weights:

1. binary weights (**3 points**);
2. raw weights (**3 points**); and
3. TF-IDF weights (**7 points**).

Use the format:

Doc # ?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Index | 1 | 2 | … |  |  |  |
| Weight | ??? | ??? |  |  |  |  |

For each form of weighting list the document vectors in the following format:

SOLUTION 4.A:

The Dictionary:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| Term | boost | digest | function | give | go | greek | help | if | immune | lactose | milk | nutrition | probiotic | sugar | system | upgrade | want | yogurt | you |
| DF | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 1 |

The document vectors: (i) Binary Weights

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Words | boost | digest | function | give | go | greek | help | if | immune | lactose | milk | nutrition | probiotic | sugar | system | upgrade | want | yogurt | you |
| DOC1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| DOC2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| DOC3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

(ii) Raw Weights

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Words | boost | digest | function | give | go | greek | help | if | immune | lactose | milk | nutrition | probiotic | sugar | system | upgrade | want | yogurt | you |
| DOC1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| DOC2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| DOC3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

(iii) TF – IDF Weights

* 1. (**15 points**) What are the hit lists for the following Boolean queries (in each case explain how you obtained them from the inverted index):

Q1. yogurt AND immune AND system (5 points)

Q2. (yogurt AND nutrition) OR (probiotic AND milk) (5 points) Q3. (sugar AND milk AND probiotic) OR nutrition (5 points)

SOLUTION 4.B:

Q1.

Yogurt: 2, 3

Immune: 2

System: 2

(yogurt and (immune and system)) => ((2,3) and (**(2) and (2)**))

=> (**(2,3) and (2)**)

=> **2**

Q2.

Yogurt: 2, 3

Nutrition: 3

Probiotic: 1,2

Milk: 1

(yogurt AND nutrition) OR (probiotic AND milk) => (**((2,3) AND (3))** OR **((1,2) AND (1))**)

=> (3 OR 1)

=> **1,3**

Q3.

Sugar: 1

Milk: 1

Probiotic: 1,2

Nutrition: 3

(sugar AND milk AND probiotic) OR nutrition => (**(1) AND (1) AND (1,2)**) OR (3)

=> (1 OR 3)

=> **1,3**

* 1. (**21 points**) Compute the similarity between Q3 from 4.B and each of the three documents from 4.A using: (i) the **lnc.ltc** scoring (5 points for each document); (ii) the Jaccard coefficient similarity (2 points for document).

SOLUTION 1.D:

(i) A picture containing text, clock

Description automatically generatedCos(Q3,D1)=

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | tf\_row | tf\_wt | df | idf | tf\_idf | Normalize |
| Milk | 1 | 1 | 1 | 0.4721 | 0.4721 | 0.5646 |
| Nutrition | 1 | 1 | 1 | 0.4771 | 0.4771 | 0.5646 |
| Probiotic | 1 | 1 | 2 | 0.1761 | 0.1761 | 0.2084 |
| Sugar | 1 | 1 | 1 | 0.4771 | 0.4771 | 0.5646 |

Document 1 (D1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | tf\_row | tf\_wt | wt | normalize | product |
| Milk | 1 | 1 | 1 | 0.5774 | 0.326 |
| Nutrition | 0 | 0 | 0 | 0 | 0 |
| Probiotic | 1 | 1 | 2 | 0.5774 | 0.1203 |
| Sugar | 1 | 1 | 1 | 0.5774 | 0.326 |

Cos(Q3,D1)= 0.7723

For Cos(Q3,D2):

Query (Q3) = (Sugar AND milk AND probiotic) OR (Nutrition)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | tf\_row | tf\_wt | df | idf | tf\_idf | normalize |
| Milk | 1 | 1 | 1 | 0.4721 | 0.4721 | 0.5646 |
| Nutrition | 1 | 1 | 1 | 0.4771 | 0.4771 | 0.5646 |
| Probiotic | 1 | 1 | 2 | 0.1761 | 0.1761 | 0.2084 |
| Sugar | 1 | 1 | 1 | 0.4771 | 0.4771 | 0.5646 |

Document 2 (D2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | tf\_row | tf\_wt | wt | normalize | product |
| Milk | 0 | 0 | 0 | 0 | 0 |
| Nutrition | 0 | 0 | 0 | 0 | 0 |
| Probiotic | 1 | 1 | 2 | 1 | 0.2084 |
| Sugar | 0 | 0 | 0 | 0 | 0 |

Cos(Q3,D2) = 0.2084

For Cos(Q3,D3):

Query (Q3) = (Sugar AND milk AND probiotic) OR (Nutrition)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | tf\_row | tf\_wt | df | idf | tf\_idf | normalize |
| Milk | 1 | 1 | 1 | 0.4721 | 0.4721 | 0.5646 |
| Nutrition | 1 | 1 | 1 | 0.4771 | 0.4771 | 0.5646 |
| Probiotic | 1 | 1 | 2 | 0.1761 | 0.1761 | 0.2084 |
| Sugar | 1 | 1 | 1 | 0.4771 | 0.4771 | 0.5646 |

Document 3 (D3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | tf\_row | tf\_wt | wt | normalize | product |
| Milk | 0 | 0 | 0 | 0 | 0 |
| Nutrition | 1 | 1 | 1 | 1 | 0.5646 |
| Probiotic | 0 | 0 | 0 | 0 | 0 |
| Sugar | 0 | 0 | 0 | 0 | 0 |

Cos(Q3, D3) = 0.5646

(ii) Jaccard Coefficients:

JC(Q3,D1)= (**Q3:** Sugar, Milk, Probiotic, Nutrition)

(**D1:** Probiotic, Help, Digest, Milk, Sugar, Lactose)

= (Q3 ∩ D1)/ (Q3 ∪ D1)

= 3/7

= 0.428

JC(Q3,D2)= (**Q3:** Sugar, Milk, Probiotic, Nutrition)

(D2: Probiotic, Yogurt, Boost, Function, Immune, System)

= (Q3 ∩ D2)/(Q3 ∪ D2)

= 1/9

= 0.111

JC(Q3,D3)= (**Q3:** Sugar, Milk, Probiotic, Nutrition)

(**D3:** If, You, Want, Give, You, Yogurt, Nutrition, Upgrade, go, Greek, Yogurt)

= (Q3 ∩ D3)/(Q3 ∪ D3)  
 =1/14  
 =0.071

**Problem 5 :** (**24 points**);

Suppose you have a collection of 5 documents, and only 10 terms are used:

Term1 Term2 Term3 Term4 Term5 Term6 Term7 Term8 Term9 Term10

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DOC1 | 0 | 3 | 2 | 4 | 0 | 5 | 0 | 0 | 4 | 2 |
| DOC2 | 3 | 0 | 1 | 4 | 3 | 0 | 0 | 5 | 1 | 6 |
| DOC3 | 6 | 0 | 5 | 1 | 2 | 0 | 2 | 5 | 0 | 7 |
| DOC4 | 1 | 8 | 0 | 2 | 0 | 1 | 6 | 0 | 2 | 1 |
| DOC5 | 2 | 7 | 0 | 0 | 0 | 3 | 0 | 2 | 3 | 0 |

List the values of the gaps for the last three terms in your index computed for this collection. (**1 point**) Encode these gaps with (i) unary codes (**3 points**); (ii) Gamma codes (**10 points**); and (iii) Delta codes (**10 points**). You are allowed to write a program to enable you computing the codes. Please add to the exam the code of the program if you chose to use one.

SOLUTION 2.I:

Index Computation:

Term 1 indexing: [2,3,4,5]

Term 2 indexing: [1,4,5]

Term 3 indexing: [1,2,3]

Term 4 indexing: [1,2,3,4]

Term 5 indexing: [2,3]

Term 6 indexing: [1,4,5]

Term 7 indexing: [3,4]

Term 8 indexing: [2,3,5]

Term 9 indexing: [1,2,4,5]

Term 10 indexing: [1,2,3,4]

Gaps for Term 8 and the unary codes for the gaps:

Gap(Difference of current value from previous value) for term 8: [2,1,2]

Unary code for n: n1’s followed by 0

Unary code for 1: 10

Unary code for 2: 110

Hence, Unary code for Gaps for Term 8: [110,10,110]

Gaps for Term 9 and the unary codes for the gaps:

Gap(Difference of current value from previous value) for term 9: [1,1,2,1]

Unary code for n: n1’s followed by 0

Unary code for 1: 10

Unary code for 2: 110

Hence, Unary code for Gaps for Term 9: [10,10,110,10]

Gaps for Term 10 and the unary codes for the gaps:

Gap(Difference of current value from previous value) for term 10: [1,1,1,1]

Unary code for n: n1’s followed by 0

Unary code for 1: 10

Unary code for 2: 110

Hence, Unary code for Gaps for Term 10: [10,10,10,10]

SOLUTION 2.II:

Gamma codes for the gaps in the posting files of Term 8:

Gap 2 :

a. Binary representation of Gap = 10

Offset = remove the leading 1 from binary representation of Gap = 0

b. Length of offset = 1

Encoding length in unary code = 10

Gamma code of Gap 2 = Concatenation of length and offset = 100

Gap 1 :

a. Binary representation of Gap = 1

Offset = remove the leading 1 from binary representation of Gap = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of Gap 1 = Concatenation of length and offset =0

Gap 2 :

a. Binary representation of Gap = 10

Offset = remove the leading 1 from binary representation of Gap = 0

b. Length of offset = 1

Encoding length in unary code = 10

Gamma code of Gap 2 = Concatenation of length and offset =100

**Gamma code for Gaps of term 8**: 100,0,100

Gamma codes for the gaps in the posting files of Term 9:

Gap 1 :

a. Binary representation of Gap = 1

Offset = remove the leading 1 from binary representation of Gap = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of Gap 1 = Concatenation of length and offset = 0

Gap 1 :

a. Binary representation of Gap = 1

Offset = remove the leading 1 from binary representation of Gap = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of Gap 1 = Concatenation of length and offset = 0

Gap 2 :

a. Binary representation of Gap = 10

Offset = remove the leading 1 from binary representation of Gap = 0

b. Length of offset = 1

Encoding length in unary code = 10

Gamma code of Gap 2 = Concatenation of length and offset =100

Gap 1 :

a. Binary representation of Gap = 1

Offset = remove the leading 1 from binary representation of Gap = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of Gap 1 = Concatenation of length and offset = 0

**Gamma code for Gaps of term 9**: 0,0,100,0

Gamma codes for the gaps in the posting files of Term 10:

Gap 1 :

a. Binary representation of Gap = 1

Offset = remove the leading 1 from binary representation of Gap = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of Gap 1 = Concatenation of length and offset = 0

Gap 1 :

a. Binary representation of Gap = 1

Offset = remove the leading 1 from binary representation of Gap = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of Gap 1 = Concatenation of length and offset = 0

Gap 1 :

a. Binary representation of Gap = 1

Offset = remove the leading 1 from binary representation of Gap = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of Gap 1 = Concatenation of length and offset = 0

Gap 1 :

a. Binary representation of Gap = 1

Offset = remove the leading 1 from binary representation of Gap = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of Gap 1 = Concatenation of length and offset = 0

**Gamma code for Gaps of term 10**: 0,0,0,0

SOLUTION 2.III:

Delta codes for the gaps in the posting files of Term 8:

Gap 2:

a. Binary representation of Gap = 10

Offset = Remove leading 1 from binary representation of Gap = 0

b. Length of Binary Representation of Gap = 2

Encode length in Gamma code :

Gamma code of 2 :

a. Binary representation of 2 = 10

Offset = remove the leading 1 from binary representation of 2 = 0

b. Length of offset = 1

Encoding length in unary code = 10

Gamma code of 2 = Concatenation of length and offset = 100

**Delta code of Gap 2** = Concatenation of length and offset = 1000

Gap 1:

a. Binary representation of Gap = 1

Offset = Remove leading 1 from binary representation of Gap = “ “

b. Length of Binary Representation of Gap = 1

Encode length in Gamma code :

Gamma code of 1 :

a. Binary representation of 1 = 1

Offset = remove the leading 1 from binary representation of 1 = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of 1 = Concatenation of length and offset = 0

**Delta code of Gap 1** = Concatenation of length and offset = 0

Gap 2:

a. Binary representation of Gap = 10

Offset = Remove leading 1 from binary representation of Gap = 0

b. Length of Binary Representation of Gap = 2

Encode length in Gamma code :

Gamma code of 2 :

a. Binary representation of 2 = 10

Offset = remove the leading 1 from binary representation of 2 = 0

b. Length of offset = 1

Encoding length in unary code = 10

Gamma code of 2 = Concatenation of length and offset = 100

**Delta code of Gap 2** = Concatenation of length and offset = 1000

**Delta codes for Gaps in posting files of term 8**: 1000,0,1000

Delta codes for the gaps in the posting files of Term 9:

Gap 1:

a. Binary representation of Gap = 1

Offset = Remove leading 1 from binary representation of Gap = “ “

b. Length of Binary Representation of Gap = 1

Encode length in Gamma code :

Gamma code of 1 :

a. Binary representation of 1 = 1

Offset = remove the leading 1 from binary representation of 1 = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of 1 = Concatenation of length and offset = 0

**Delta code of Gap 1** = Concatenation of length and offset = 0

Gap 1:

a. Binary representation of Gap = 1

Offset = Remove leading 1 from binary representation of Gap = “ “

b. Length of Binary Representation of Gap = 1

Encode length in Gamma code :

Gamma code of 1 :

a. Binary representation of 1 = 1

Offset = remove the leading 1 from binary representation of 1 = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of 1 = Concatenation of length and offset = 0

**Delta code of Gap 1** = Concatenation of length and offset = 0

Gap 2:

a. Binary representation of Gap = 10

Offset = Remove leading 1 from binary representation of Gap = 0

b. Length of Binary Representation of Gap = 2

Encode length in Gamma code :

Gamma code of 2 :

a. Binary representation of 2 = 10

Offset = remove the leading 1 from binary representation of 2 = 0

b. Length of offset = 1

Encoding length in unary code = 10

Gamma code of 2 = Concatenation of length and offset = 100

**Delta code of Gap 2** = Concatenation of length and offset = 1000

Gap 1:

a. Binary representation of Gap = 1

Offset = Remove leading 1 from binary representation of Gap = “ “

b. Length of Binary Representation of Gap = 1

Encode length in Gamma code :

Gamma code of 1 :

a. Binary representation of 1 = 1

Offset = remove the leading 1 from binary representation of 1 = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of 1 = Concatenation of length and offset = 0

**Delta code of Gap 1** = Concatenation of length and offset = 0

**Delta codes for Gaps in posting files of term 9**: 0,0,1000,0

Delta codes for the gaps in the posting files of Term 10:

Gap 1:

a. Binary representation of Gap = 1

Offset = Remove leading 1 from binary representation of Gap = “ “

b. Length of Binary Representation of Gap = 1

Encode length in Gamma code :

Gamma code of 1 :

a. Binary representation of 1 = 1

Offset = remove the leading 1 from binary representation of 1 = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of 1 = Concatenation of length and offset = 0

**Delta code of Gap 1** = Concatenation of length and offset = 0

Gap 1:

a. Binary representation of Gap = 1

Offset = Remove leading 1 from binary representation of Gap = “ “

b. Length of Binary Representation of Gap = 1

Encode length in Gamma code :

Gamma code of 1 :

a. Binary representation of 1 = 1

Offset = remove the leading 1 from binary representation of 1 = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of 1 = Concatenation of length and offset = 0

**Delta code of Gap 1** = Concatenation of length and offset = 0

Gap 1:

a. Binary representation of Gap = 1

Offset = Remove leading 1 from binary representation of Gap = “ “

b. Length of Binary Representation of Gap = 1

Encode length in Gamma code :

Gamma code of 1 :

a. Binary representation of 1 = 1

Offset = remove the leading 1 from binary representation of 1 = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of 1 = Concatenation of length and offset = 0

**Delta code of Gap 1** = Concatenation of length and offset = 0

Gap 1:

a. Binary representation of Gap = 1

Offset = Remove leading 1 from binary representation of Gap = “ “

b. Length of Binary Representation of Gap = 1

Encode length in Gamma code :

Gamma code of 1 :

a. Binary representation of 1 = 1

Offset = remove the leading 1 from binary representation of 1 = “ ”

b. Length of offset = 0

Encoding length in unary code = 0

Gamma code of 1 = Concatenation of length and offset = 0

**Delta code of Gap 1** = Concatenation of length and offset = 0

**Delta codes for Gaps in posting files of term 10**: 0,0,0,0