**ΕΡΓΑΣΙΑ 2**

**ΣΤΟ ΜΑΘΗΜΑ: ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΜΗΧΑΝΕΣ ΑΝΑΖΗΤΗΣΗΣ**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Dec | Bin | Offset | Unary | γ-code | Length δ-code | δ-code | Variable byte code |
| 44 | 101100 | 01100 | 111110 | 11111001100 | 110001 | 1100101100 | 10101100 |
| 15 | 1111 | 111 | 1110 | 1110111 | 101 | 101111 | 10001111 |
| 21 | 10101 | 0101 | 11110 | 111100101 | 11000 | 110000101 | 10010101 |
| 5 | 101 | 01 | 110 | 11001 | 100 | 10001 | 10000101 |
| 14 | 1110 | 110 | 1110 | 1110110 | 101 | 101110 | 10001110 |
| 201 | 11001001 | 1001001 | 11111110 | 111111101001001 | 11011 | 110111001001 | 00000001  11001001 |
| 1 | 1 | 1 | 0 | 0 | - | - | 10000001 |

Gap list <44,15,21,5,14,201,1>

Έχουμε ως δεδομένο ότι η ασυμπίεστη εκδοχή της λίστας είναι 224bits.

Με γ-code έχουμε συμπίεση 11+7+9+5+7+15+1 = 55 bits δηλαδή 24,55% επί του αρχικού.

Με δ-code δεν έχουμε συμπίεση γιατί χάθηκε ένα bit.

Με variable byte code έχουμε συμπίεση 8+8+8+8+8+16+8 = 64 bits δηλαδή 28,57% επί του αρχικού.

1. A) Το πρώτο bit της οχτάδας είναι header άρα μας μένουν 7bit. Με 7bit μπορούμε να έχουμε 2^7-1 = 127.

Β) Το πρώτο bit κάθε οχτάδας είναι header άρα μας μένουν 7 + 7 = 14bit. Με 14bit μπορούμε να έχουμε 2^14-1 = 16383.

Γ) Στην gap list που μας δίνεται έχουμε 11 αριθμούς εκ των οποίων οι 2 είναι μεγαλύτεροι του 127 και όλοι οι υπόλοιποι είναι μικρότεροι του 16383, άρα

θα έχουμε 9\*1+2\*2 = 13 bytes.

1. Κάθε φορά κοιτάμε το unary για να βγάλουμε τους αριθμούς μας.

Αρχικά θα έχουμε 1110,001 δηλαδή 001=1001=9

Ακολουθούμε την ιδία διαδικασία και για τα επόμενα

110,10 = 110 = 6

10,1 = 11 = 3

111110,11011 = 111011 = 59

110,11 = 111 = 7

Οπότε προκύπτει η gap list με στοιχειά <9,6,3,59,7> και η posting list με στοιχεία <9,15,18,77,84>.

1. **Υλοποίηση σε Java.**

**public** **class** main {

**public** **static** **void** main(String[] args){

//Ορισμός αριμθού για encoding

**int** num = 214577;

//δημιουργία λίστα η οποία θα περιέχει ως στοιχεία integer που θα εμφανίζονται ως 8bit στο τελος

ArrayList<Integer> list = **new** ArrayList<>();

//διαιρεση num με 128 ωστε να παρουμε 1 byte (128 "χανουμε" απο τον header) και πρόσθεση πρώτου αριθμού αυξημένου με 128(c=1 header)

**int** f;

f=(num % 128);

list.add(f+128);

num = Math.*floorDiv*(num, 128);

//το ιδιο απλά χωρίς αυξηση με 128 (c=0 header)

**if**(num>0)

**while**(**true**)

{

f=(num % 128);

list.add(f);

**if**(num < 128)

**break**;

num = Math.*floorDiv*(num, 128);

}

//αντιστροφη λιστας ωστε να φαινονται με την σωστη σειρα

Collections.*reverse*(list);

// int->binary και print

**for**(**int** item:list)

{

String result = Integer.*toBinaryString*(item);

String padding = String.*format*("%8s", result).replace(' ', '0');

System.***out***.println(padding);

}

}

}

1. Για κάθε Doc θα υπολογίσουμε ξεχωριστά τα tf-wt, wt και τα normalized.

**Doc1**

tf-wtcar =1 + log1027 = 2,43

tf-wtauto = 1 + log103 = 1,47

tf-wtinsurance = 1 + log100 = 0

tf-wtbest = 1 + log1014 = 2,14

wtcar,doc1= 2,43 \* 1,65 = 4,00

wtauto,doc1= 1,47 \* 2,08= 3,05

wtinsurance,doc1 = 0 \* 1,62= 0

wtbest,doc1= 2,14 \* 1,5= 3,21

length =

normalizedcar = wtcar,doc1/ 5,96 = 4,00/5,96 = 0,67

normalizedauto= wtauto,doc1/ 5,96 = 3,05/5,96 = 0,51

normalizedinsurance= wtinsurance,doc1 / 5,96 = 0 / 5,96 = 0

normalizedbest= wtbest,doc1/ 5,96 = 3,21/5,96 = 0,53

**Doc2**

tf-wtcar = 1+log104=1,6

tf-wtauto = 1+log1033=2,5

tf-wtinsurance = 1+ log1033=2,5

tf-wtbest = 1 + log100 = 0

wtcar,doc2 = 1,6 \* 1,65 = 2,64

wtauto,doc2 = 2,5 \* 2,08 = 5,20

wtinsurance,doc2 = 2,5 \* 1,62 = 4,05

wtbest,doc2 = 0 \* 1,5=0

length=

normalizedcar=wtcar,doc2/ 7,10 =2,64/7,10 = 0,37

normalizedauto= wtauto,doc2/ 7,10 =5,20/7,10 = 0,73

normalizedinsurance= wtinsurance,doc2/ 7,10 =4,05/7,10 = 0,57

normalizedbest= wtbest,doc2 / 7,10 = 0 / 7,10 = 0

**Doc3**

tf-wtcar=1+log1024 = 2,38

tf-wtauto=1 + log100 = 0

tf-wtinsurance=1+log1029 =2,46

tf-wtbest=1+log1017=2,2

wtcar,doc3= 2,38 \* 1,65 = 3,92

wtauto,doc3=0 \* 2,08 =0

wtinsurance,doc3=2,46 \* 1,62 = 3,98

wtbest,doc3= 2,2 \* 1,5 =3,30

length=

normalizedcar =wtcar,doc3/ 6,48 =3,92/6,48 = 0,60

normalizedauto = wtauto,doc3 / 6,48 = 0 / 6,48 = 0

normalizedinsurance = wtinsurance,doc3/ 6,48 =3,98/6,48 = 0,61

normalizedbest = wtbest,doc3/ 6,48 =3,30/6,48 = 0,50

Συνοψίζοντας όλα τα παραπάνω θα προκύψει ο εξής πινάκας:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Doc1** | | **Doc2** | | **Doc3** | |
| **wt** | **normal** | **wt** | **normal** | **wt** | **normal** |
| **car** | 4,00 | 0,67 | 2,64 | 0,37 | 3,92 | 0,60 |
| **auto** | 3,05 | 0,51 | 5,20 | 0,73 | 0 | 0 |
| **insurance** | 0 | 0 | 4,05 | 0,57 | 3,98 | 0,61 |
| **best** | 3,21 | 0,53 | 0 | 0 | 3,30 | 0,50 |

1. Α) Έχουμε **P = q\*Doc** και q = [1, 0, 1, 0]. Άρα θα προκύψουν οι εξής πράξεις:

**Doc 1**

Pcar,Doc1 = 1 \* 0,67 = 0,67

Pauto,Doc1 = 0 \* 0,51 = 0

Pinsurance,Doc1 = 1 \* 0 = 0

Pbest,Doc1 = 0 \* 0,53 = 0

**Doc 2**

Pcar,Doc2 = 1 \* 0,37 = 0,37

Pauto,Doc2 = 0 \* 0,73 = 0

Pinsurance,Doc2 = 1 \* 0,57 = 0,57

Pbest,Doc2 = 0 \* 0= 0

**Doc 3**

Pcar,Doc3 = 1 \* 0,60 = 0,60

Pauto,Doc3 = 0 \* 0 = 0

Pinsurance,Doc3 = 1 \* 0,61 = 0,61

Pbest,Doc3 = 0 \* 0,55 = 0

**ScoreDoc1 = 0,67 + 0 + 0 + 0 = 0,67**

**ScoreDoc2­ = 0,37 + 0 + 0,57 + 0 = 0,94**

**ScoreDoc3 = 0,60 + 0 + 0,61 + 0 = 1,21**

Άρα η σειρά που προκύπτει είναι: doc3, doc2, doc1.

B) Το idf ‘’car insurance’’ έχει 1\*1,65=1,65 και 1\*1,62 αντίστοιχα.

Και προκύπτουν οι εξής πράξεις:

normalizedQuerycar =

normalizedQueryauto = 0

normalizedQueryinsurance =

normalizedQuerybest = 0

**Doc1**

P1car = normalizedcar\* Doc1car­ = 0,71 \* 0,67 = 0,46

P1auto = normalizedauto\* Doc1auto = 0 \* 0,51 = 0

P1insurance = normalizedinsurance\* Doc1insurance  = 0,70 \* 0 = 0

P1best = normalizedbest\* Doc1best­ = 0 \* 0,53 = 0

**Doc2**

P2car = normalizedcar\* Doc2car­ = 0,71 \* 0,37 = 0,26

P2auto = normalizedauto\* Doc2auto = 0 \* 0,73 = 0

P2insurance = normalizedinsurance\* Doc2insurance  = 0,70 \* 0,57= 0,39

P2best = normalizedbest\* Doc2best­ = 0 \* 0 = 0

**Doc3**

P3car = normalizedcar\* Doc3car­ = 0,71 \* 0,60 = 0,42

P3auto = normalizedauto\* Doc3auto = 0 \* 0 = 0

P3insurance = normalizedinsurance\* Doc3insurance  = 0,70 \* 0,61 = 0,42

P3best = normalizedbest\* Doc3best­ = 0 \* 0,50 = 0

**ScoreDoc1 = 0,46 + 0 + 0 + 0 = 0,46**

**ScoreDoc2­ = 0,26 + 0 + 0,39 + 0 = 0,65**

**ScoreDoc3 = 0,42 + 0 + 0,42 + 0 = 0,84**

Άρα η σειρά που προκύπτει είναι: doc3, doc2, doc1.

**ΑΠΟ ΤΟΝ ΦΟΙΤΗΤΗ: Τσιαούση Σταύρο (dai17173)**