**Στέργιος Τσάνταλης (iis21125)**

1. (α)

|  |  |  |  |
| --- | --- | --- | --- |
| Doc1 | Doc2 |  |  |

οι =>

|  |  |  |  |
| --- | --- | --- | --- |
| Doc1 | Doc2 | Doc3 | Doc4 |

πωλήσεις =>

|  |  |  |  |
| --- | --- | --- | --- |
| Doc1 | Doc4 |  |  |

νεόκτιστων =>

|  |  |  |  |
| --- | --- | --- | --- |
| Doc1 | Doc2 | Doc3 | Doc4 |

|  |  |  |  |
| --- | --- | --- | --- |
| Doc1 |  |  |  |

διαμερισμάτων =>

ξεπέρασαν =>

|  |  |  |  |
| --- | --- | --- | --- |
| Doc1 |  |  |  |

τις =>

|  |  |  |  |
| --- | --- | --- | --- |
| Doc1 |  |  |  |

προβλέψεις =>

|  |  |  |  |
| --- | --- | --- | --- |
| Doc2 |  |  |  |

παρουσιάζουν =>

|  |  |  |  |
| --- | --- | --- | --- |
| Doc2 |  |  |  |

άνοδο =>

|  |  |  |  |
| --- | --- | --- | --- |
| Doc2 | Doc3 | Doc4 |  |

τον =>

|  |  |  |  |
| --- | --- | --- | --- |
| Doc2 | Doc3 | Doc4 |  |

Ιούλιο =>

|  |  |  |  |
| --- | --- | --- | --- |
| Doc3 | Doc4 |  |  |

αύξηση =>

|  |  |  |  |
| --- | --- | --- | --- |
| Doc3 | Do4 |  |  |

στις =>

|  |  |  |  |
| --- | --- | --- | --- |
| Doc4 |  |  |  |

υπάρχει =>

(β) (i) Doc3, Doc4

(ii) Δεν επιστρέφει κάποιο έγγραφο.

1. Χρησιμοποιώντας τη συντηρητική εκτίμηση του μήκους των ενωμένων postings lists, η προτεινόμενη σειρά είναι: (υπολογιστής OR μολύβι) (49,000) AND (λάμπα OR κουρτίνα) (55,000) AND (τραπέζι OR καρέκλα) (57,000). Ωστόσο, ανάλογα με την πραγματική κατανομή των postings, (λάμπα OR κουρτίνα) μπορεί να είναι μακρύτερα από (τραπέζι OR καρέκλα) επειδή τα δύο συστατικά του πρώτου είναι πιο ασύμμετρα.

Η λύση του S.Singh

Χρόνος επεξεργασίας: (i) (τραπέζι OR καρέκλα) = Ο(25,000+32,000)=Ο(57,000)

(ii) (λάμπα OR κουρτίνα) = Ο(5,000+50,000) = Ο(55,000)

(iii) (υπολογιστής OR μολύβι) = Ο(18,000+31,000) = Ο(49,000)

Σειρά επεξεργασίας:

Α. Process (i),(ii),(iii) με οποιαδήποτε σειρά ως τα πρώτα 3 βήματα (συνολικός χρόνος για αυτά τα βήματα είναι Ο(57,000+55,000+49,000) σε κάθε περίπτωση).

Β. Merge (ii) AND (iii) = (iv): Στην περίπτωση του τελεστή AND, η πολυπλοκότητα της συγχωνευόμενης postings list εξαρτάται από το μήκος της πιο σύντομης postings list. Επομένως, όσο πιο σύντομη με τα μικρότερα postings είναι η λίστα, τόσο μικρότερος είναι ο χρόνος που δαπανάται. Ο λόγος για την επιλογή του (ii) αντί του (i) είναι ότι η output list (iv) είναι πιο πιθανό να είναι μικρότερη αν επιλεγεί το (ii).

C. Merge (iv) AND (i): Αυτή είναι η μόνη διαδικασία συγχώνευσης που έχει απομείνει.

1. (α) Merge(X,Y)=Result1 με Ο(100+105)=Ο(205) και μήκος 100.

Merge(Result1,Z) με Ο(100+110)=Ο(210).

Άρα το συνολικό κόστος είναι 205+210=415.

(β) Άρα ξέρουμε ότι Ο(Χ+Υ)=Ο(100) και Ο(Χ+Ζ) =Ο(0) επομένως το κόστος θα είναι 100+0=100.

1. (α) Ταιριάζουν και τα τρία έγγραφα.

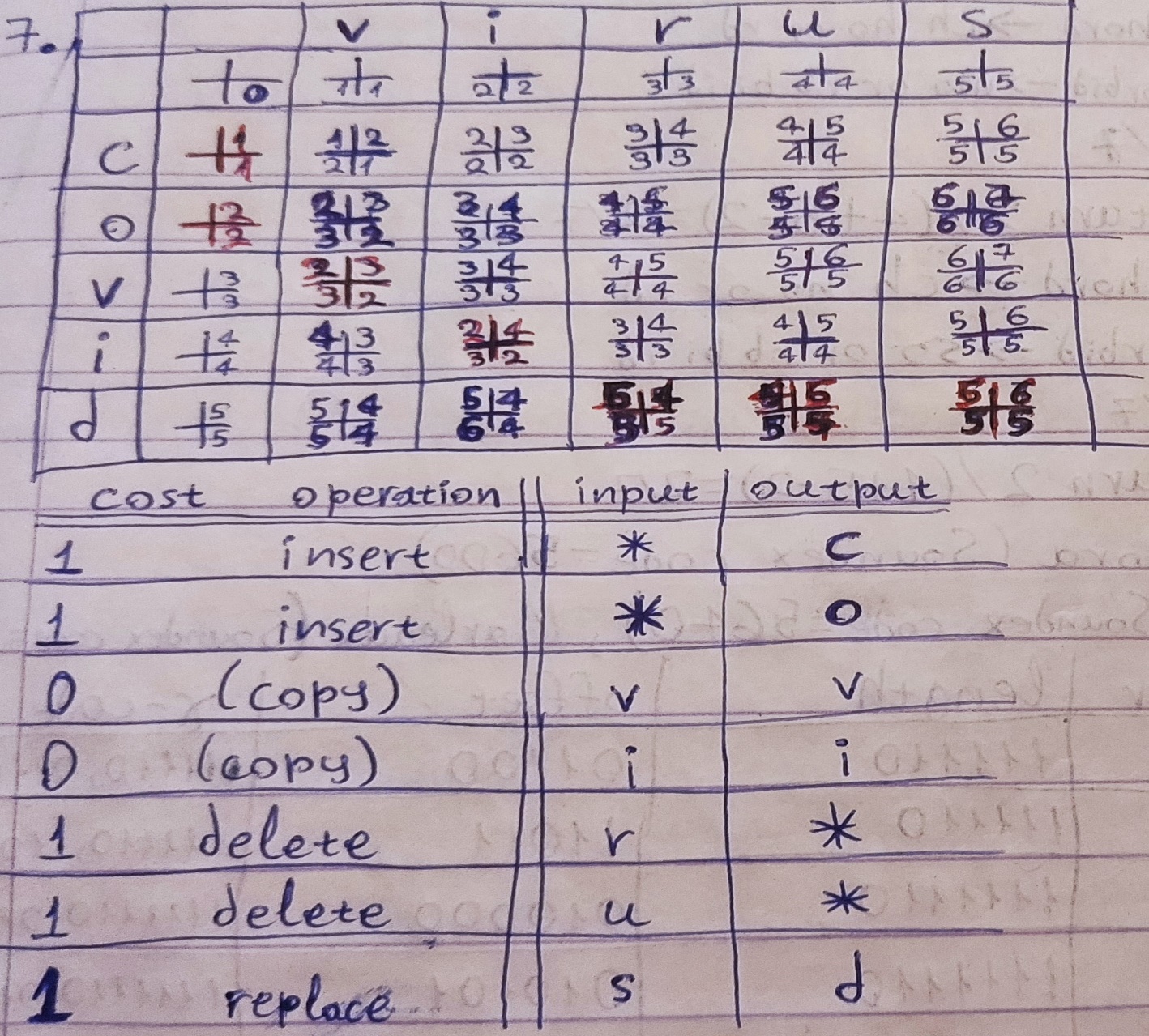
(β) Ταιριάζει το 4.

1. (α) Εφαρμόζοντας Merge στις postings lists, θα γίνονται συγκρίσεις εκτός εάν τελειώσει κάποια από αυτές, δηλαδή μέχρι να φτάσουμε στο 47 στην πρώτη postings list, μετά το οποίο τελειώνει η δεύτερη postings list και δεν χρειάζεται να συνεχιστεί η διαδικασία. Αριθμός συγκρίσεων = 11

(β) Χρησιμοποιώντας skip pointers μήκους 4 για την μεγαλύτερη λίστα και μήκους 1 για την μικρότερη λίστα, θα γίνουν οι ακόλουθες συγκρίσεις:

1. 4&47 2. 14&47 3.22&47 4. 120&47 5. 81&47 6. 47&47 Αριθμός συγκρίσεων = 6

1. Ένας. (Εάν δεν υπήρχε το σύμβολο $, θα μπορούσαμε να έχουμε πολλούς πρωτότυπους όρους λεξιλογίου που προκύπτουν από εναλλαγές permuterm όρου λεξιλογίου, π.χ. γραφή και φραγή.



1. Query: chord -> ch ho or rd

Term: border -> bo or rd de er

JC = return 2/(4+5-4) = 2/5

Query: chord -> ch ho or rd

Term: lord -> lo or rd

JC = return 2/(4+3-4) = 2/3

Query: chord -> ch ho or rd

Term: morbid -> mo or rb bi id

JC = return 2/(4+5-2) = 2/7

Query: chord -> ch ho or rd

Term: sorbid -> so or rb bi id

JC = return 2/(4+5-2) = 2/7

1. Mary,Nora (Soundex code = 5600)
2. Marlo (Soundex code = 5640), Marleaux (Soundex code = 5642)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | length | offset | γ-code | δ-code |
| 44 | 111110 | 01100 | 11111001100 | 11101001100 |
| 59 | 111110 | 11011 | 11111011011 | 11101011011 |
| 80 | 1111110 | 010000 | 1111110010000 | 111011010000 |
| 85 | 1111110 | 010101 | 1111110010101 | 111011010101 |
| 99 | 1111110 | 100011 | 1111110100011 | 111011100011 |
| 300 | 111111110 | 00101100 | 11111111000101100 | 1111000100101100 |
| 301 | 111111110 | 00101101 | 11111111000101101 | 1111000100101101 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Numbers | 44 | 59 | 80 | 85 | 99 | 300 | 301 |
| Gaps |  | 15 | 21 | 5 | 14 | 201 | 1 |
| VB code | 00101100 | 10000011 | 10010101 | 10000101 | 10001110 | 00000001  11001001 | 10000001 |

1. (α) 127 (β)16384 (γ) 13 bytes
2. 1110001110101011111101101111011

Gaps list 9 6 3 119 7

Postings list 9 15 18 137 144

1. Για το car:

Doc1: w = log10(1+27) \* 1.65 = 2.39

Doc2: w = log10(1+4) \* 1.65 = 1.15

Doc3: w = log10(1+24) \* 1.65 = 2.31

Για το auto:

Doc1: w = log10(1+3) \* 2.08 = 1.25

Doc2: w = log10(1+33) \* 2.08 = 3.18

Do3: w = 0

Για το insurance:

Doc1: w = 0

Doc2: w = log10(1+33) \* 1.62 = 2.49

Doc3: w = log10(1+29) \* 1.62 = 2.39

Για το best:

Doc1: w = log10(1+14) \* 1.5 = 1.76

Doc2: w = 0

Doc3: w = log10(1+17) \* 1.5 = 1.88

|  |  |  |  |
| --- | --- | --- | --- |
| Term | Doc1 | Doc2 | Doc3 |
| Car | 2.39 | 1.15 | 2.31 |
| Auto | 1.25 | 3.18 | 0 |
| Insurance | 0 | 2.49 | 2.39 |
| Best | 1.76 | 0 | 1.88 |

Doc1: sas =√ (2.392+1.252+02+1.762)≈3.22

Doc2: sas =√ (1.152+3.182+2.492+02)≈3.22

Doc3: sas =√ (2.312+02+2.392+1.882)≈3.22

Κάνοντας την διαιρέσεις δημιουργούνται τα ακόλουθα διανύσματα:

Doc1 = (0.742,0.388,0,0.546)

Doc2 = (0.274,0.758,0.594,0)

Doc3 = (0.604,0,0.625,0.492)

1. (i)

|  |  |  |  |
| --- | --- | --- | --- |
| Term | Doc1 | Doc2 | Doc3 |
| Car | 0.742 | 0.274 | 0.604 |
| Auto | 0.388 | 0.758 | 0 |
| Insurance | 0 | 0.594 | 0.625 |
| Best | 0.546 | 0 | 0.492 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Term | Query(q) | | | | | Product |
|  | tf | W(t,q) | Doc1 | Doc2 | Doc3 |  |
| Car | 1 | 1 | 0.742 | 0.274 | 0.604 | 1.62 |
| Auto | 0 | 0 | 0 | 0 | 0 | 0 |
| Insurance | 1 | 1 | 0 | 0.594 | 0.625 | 1.219 |
| Best | 0 | 0 | 0 | 0 | 0 | 0 |

Score(q,Doc1) = 0.742

Score(q,Doc2) = 0.274+0.594 = 0.868

Score(q,Doc3) = 0.604+0.625 = 1.229

Ranking = Doc3, Doc2, Doc1

(ii)

|  |  |  |
| --- | --- | --- |
| Term | Query(q) | |
|  | Idf | W(t,q) |
| Car | 1.65 | 0.478 |
| Auto | 2.08 | 0.602 |
| Insurance | 1.62 | 0.47 |
| Best | 1.5 | 0.43 |

Score(q,Doc1) = 0.686

Score(q,Doc2) = 0.797

Score(q,Doc3) = 0.781

Ranking = Doc2, Doc3, Doc1