Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών
ΗΥ463 - Συστήματα Ανάκτησης Πληροφοριών
2007-2008 Εαρινό Εξάμηνο

Λύσεις 2η Σειρά Ασκήσεων
(Μοντέλα Ανάκτησης)

Άσκηση 1 (2 βαθμοί)
(α) Δώστε τη διανυσματική παράσταση των εγγράφων $d_1, \ldots, d_5$ με βάρη TF-IDF. Θεωρείστε ότι η θέση της κάθε λέξης στα διανύσματα γίνεται κατά αλφαβητική σειρά.
(β) Δώστε την απάντηση που θά έχει η κάθε επερώτηση $q_1, \ldots, q_4$ βάσει του διανυσματικού μοντέλου.
(γ) Σχεδιάστε τη μορφή που θα έχει το ανεστραμμένο ευρετήριο για τη συλλογή D.

Documents D
d_1 : "a b"
d_2 : "a b a b"
d_3 : "a b a b c"
d_4 : "a b c"
d_5 : "a a c"

Queries Q
q_1 : "a b"
q_2 : "a"
q_3 : "c"
q_4 : "a c"

ΛΥΣΗ

(α) Term Occurrence Table

	a	b	c	$MAX_k\{FREQ_{ij}\}$
d_1	1	1	0	1
d_2	2	2	0	2
d_3	2	2	1	2
d_4	1	1	1	1
d_5	2	0	1	2
df	5	4	3	
idf	5/5	5/4	5/3	

- $FREQ_{ij}$ = το πλήθος των εμφανίσεων του όρου i στο έγγραφο j
- N=5
- IDF = N / DF
- $MAX_k\{FREQ_{ij}\}$ = συχνότητα της λέξης με τη μέγιστη συχνότητα στο κείμενο

Term Weight Table

	a	b	c	$MAX_k\{FREQ_ij\}$
d_1	1/1*5/5	1/1*5/4	0	1
d_2	2/2*5/5	2/2*5/4	0	2
d_3	2/2*5/5	2/2*5/4	1/2*5/3	2
d_4	1/1*5/5	1/1*5/4	1/1*5/3	1
d_5	2/2*5/5	0	1/2*5/3	2
df	5	4	3	
idf	5/5	5/4	5/3	

- $TF_{ij} = FREQ_{ij}/MAX_k\{FREQ_{ij}\}$
- $V_{ij} = TF_{ij} * IDF_i$

Οι διανυσματικές παραστάσεις των εγγράφων είναι :
$V_1 = \{1, 1.25, 0\}$, $|V_1| = 2.5625$
$V_2 = \{1, 1.25, 0\}$, $|V_2| = 2.5625$
$V_3 = \{1, 1.25, 0.83\}$, $|V_3| = 3.2514$
$V_4 = \{1, 1.25, 1.66\}$, $|V_4| = 5.3181$
$V_5 = \{1, 0, 0.83\}$, $|V_5| = 1.6889$

(β)

	a	b	c
q_1	1/1*5/5	1/1*5/4	0
q_2	1/1*5/5	0	0
q_3	0	0	1/1*5/3
q_4	1/1*5/5	0	1/1*5/3
df	5/5	5/4	5/3

$q_1 = \{1, 1.25, 0\}$, $|q_1| = 2.5625$
$q_2 = \{1, 0, 0\}$, $|q_2| = 1$
$q_3 = \{0, 0, 1.66\}$, $|q_3| = 2.7556$
$q_4 = \{1, 0, 1.66\}$, $|q_4| = 3.7556$

Για κάθε έγγραφο D_j υπολογίζουμε το μέτρο ομοιότητας συνημίτονου
$R(d_j, q) = CosSim(d_j, q) = \frac{\vec{d_j} \bullet \vec{q}}{|\vec{d_j}|x|\vec{q}|} = \frac{\sum_{i=1}^{t} w_{i,j} x w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} x \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$
Για την επερώτηση q_1 έχουμε:
$V_1 * q_1 = 1*1+1.25*1.25+0*0 = 1+1.5625 = 2.5625$
$V_2 * q_1 = 1*1+1.25*1.25+0*0 = 2.5625$
$V_3 * q_1 = 1*1+1.25*1.25+0.83*0 = 2.5625$
$V_4 * q_1 = 1*1+1.25*1.25+1.66*0 = 2.5625$
$V_5 * q_1 = 1*1+0*1.25+0.83*0 = 1$

$R(d_1, q_1) = \frac{2.5625}{\sqrt{2.5625*2.5625}} = 1$
$R(d_2, q_1) = \frac{2.5625}{\sqrt{2.5625*2.5625}} = 1$
$R(d_3, q_1) = \frac{2.5625}{\sqrt{3.2514*2.5625}} = 0.887$
$R(d_4, q_1) = \frac{2.5625}{\sqrt{5.3181*2.5625}} = 0.694$
$R(d_5, q_1) = \frac{1}{\sqrt{1.6889*2.5625}} = 0.48$

Με βάση το διανυσματικό μοντέλο η διάταξη των εγγράφων για την επερώτηση q_1 είναι : $< \{d_1, d_2\}, d_3, d_4, d_5 >$
Τα έγγραφα d_1, d_2 βρίσκονται στην 1η θέση επειδή περιέχουν όλους τους όρους της επερώτησης και μόνο αυτούς. Στη 2η θέση βρίσκονται τα έγγραφα d_3, d_4 γιατί παρόλο που περιέχουν όλους τους όρους της επερώτησης περιέχουν και τον όρο "c",ο οποίος επηρεάζει το βάρος τους.

Για την επερώτηση q_2 έχουμε:
$V_1 * q_2 = 1*1+0+0 = 1$
$V_2 * q_2 = 1*1+0+0 = 1$
$V_3 * q_2 = 1*1+0+0 = 1$
$V_4 * q_2 = 1*1+0+0 = 1$
$V_5 * q_2 = 1*1+0+0 = 1$

$R(d_1, q_2) = \frac{1}{\sqrt{2.5625*1}} = 0.625$
$R(d_2, q_2) = \frac{1}{\sqrt{2.5625*1}} = 0.625$

$R(d_3, q_2) = \frac{1}{\sqrt{3.2514*1}} = 0.555$
$R(d_4, q_2) = \frac{1}{\sqrt{5.3181*1}} = 0.433$
$R(d_5, q_2) = \frac{1}{\sqrt{1.6889*1}} = 0.769$

Με βάση το διανυσματικό μοντέλο η διάταξη των εγγράφων για την επερώτηση q_2 είναι : $< d_5, \{d_1, d_2\}, d_3, d_4 >$

Για την επερώτηση q_3 έχουμε:
$V_1 * q_3 = 0$
$V_2 * q_3 = 0$
$V_3 * q_3 = 0$+0+0.83*1.66 = 1.3778
$V_4 * q_3 = 0$+0+1.66*1.66 = 2.7556
$V_5 * q_3 = 0$+0+0.83*1.66 = 1.3778

$R(d_1, q_3) = 0$
$R(d_2, q_3) = 0$
$R(d_3, q_3) = \frac{1.3778}{\sqrt{3.2514*2.7556}} = 0.46$
$R(d_4, q_3) = \frac{2.7556}{\sqrt{5.3181*2.7556}} = 0.718$
$R(d_5, q_3) = \frac{1.3778}{\sqrt{1.6889*2.7556}} = 0.638$

Με βάση το διανυσματικό μοντέλο η διάταξη των εγγράφων για την επερώτηση q_3 είναι : $< d_4, d_5, d_3 >$

Για την επερώτηση q_4 έχουμε:
$V_1 * q_4 = 1$*1+0+0 = 1
$V_2 * q_4 = 1$*1+0+0 = 1
$V_3 * q_4 = 1$*1+0+0.83*1.66 = 2.3778
$V_4 * q_4 = 1$*1+0+1.66*1.66 = 3.7556
$V_5 * q_4 = 1$*1+0+0.83*1.66 = 2.3778

$R(d_1, q_4) = \frac{1}{\sqrt{2.5625*3.7556}} = 0.322$
$R(d_2, q_4) = \frac{1}{\sqrt{2.5625*3.7556}} = 0.322$
$R(d_3, q_4) = \frac{2.3778}{\sqrt{3.2514*3.7556}} = 0.68$
$R(d_4, q_4) = \frac{3.7556}{\sqrt{5.3181*3.7556}} = 0.84$
$R(d_5, q_4) = \frac{2.3778}{\sqrt{1.6889*3.7556}} = 0.944$

Με βάση το διανυσματικό μοντέλο η διάταξη των εγγράφων για την επερώτηση q_4 είναι : $< d_5, d_4, d_3, \{d_1, d_2\} >$

(γ) Ανεστραμμένο ευρετήριο
Μία μορφή του ανεστραμμένου ευρετηρίου στο οποίο εμφανίζονται μόνο οι θέσεις των όρων είναι :

Term	$< DocumentFrequency, (Document; Position) >$
a	$< 5, (d_1; 1), (d_2; 1), (d_2; 3), (d_3; 1), (d_3; 3), (d_4; 1), (d_5; 1), (d_5; 2) >$
b	$< 4, (d_1; 2), (d_2; 2), (d_2; 4), (d_3; 2), (d_3; 4), (d_4; 2) >$
c	$< 3, (d_3; 5), (d_4; 3), (d_5; 3) >$

Άσκηση 2 (2 βαθμοί)
Έστω οτι έχουμε ένα μοντέλο ανάκτησης το οποίο βλέπει τα έγγραφα και τις επερωτήσεις ως σύνολα όρων. Συγκρίνετε τις ακόλουθες συναρτήσεις διαβάθμισης

$$R_1(d, q) = \frac{|d \cap q|}{|q|}$$

$$
\begin{aligned}
R_2(d,q) &= \frac{|d \cap q|}{|d|} \\
R_3(d,q) &= \frac{|d \cap q|}{|d \cup q|} \\
R_4(d,q) &= \frac{|d \cap q|}{|d| + |q|} \\
R_5(d,q) &= |d \cap q|
\end{aligned}
$$

Μπορείτε να στηρίξετε την απάντησή σας παραθέτοντας παραδείγματα.

ΛΥΣΗ

Σαν παράδειγμα θεωρούμε τα έγγραφα της άσκησης 1 και την επερώτηση q="a b".

	R_1	R_2	R_3	R_4	R_5
d_1="a b"	2/2=1	2/2=1	2/2=1	2/4=0.5	2
d_2="a b a b"	2/2=1	2/4=0.5	2/2=1	2/6=0.333	2
d_3="a b a b c"	2/2=1	2/5=0.4	2/3=0.667	2/7=0.285	2
d_4="a b c"	2/2=1	2/3=0.667	2/3=0.667	2/5=0.4	2
d_5="a a c"	1/2=0.5	1/3=0.333	1/3=0.333	1/5=0.2	2
Διάταξη εγγράφων	$\{d_1, d_2, d_3, d_4\}, d_5$	d_1, d_4, d_2, d_3, d_5	$\{d_1, d_2\}, \{d_3, d_4\}, d_5$	d_1, d_4, d_2, d_3, d_5	$\{d_1, d_2, d_3, d_4, d_5\}$

R_1

Το q είναι πάντα το ίδιο επομένως δεν λαμβάνονται υπόψη οι λέξεις του κάθε εγγράφου που δεν ταιριάζουν με την επερώτηση. Έτσι τα έγγραφα που εκτός από τις λέξεις της επερώτησης περιέχουν και άλλες θα έχουν την ίδια διαβάθμιση με τα έγγραφα που περιέχουν μόνο τους όρους της επερώτησης.

R_2

Ο παρανομαστής είναι πάντα διαφορετικός άρα λαμβάνει υπόψη το μέγεθος του κάθε αρχείου και κατ΄ επέκταση το ποσοστό του εγγράφου στο οποίο δεν έχουμε ταίριασμα. Οπότε θα έχουμε διαφορετική διαβάθμιση για τα έγγραφα που περιέχουν μόνο τους όρους της επερώτησης και αυτά που περιέχουν και άλλους. Όμως, αν πάρουμε ένα έγγραφο d_6="a", το οποίο ο μόνος όρος που περιέχει είναι όρος της επερώτησης, τότε $R_2(d_6, q) = 1/1 = 1$. Δηλαδή βλέπουμε ότι έχει μεγαλύτερη συνάφεια από το d_4 το οποίο περιέχει και τους δύο όρους της επερώτησης. Και μάλιστα έχει συνάφεια 1, όπως έχει και το έγγραφο d_2 που περιέχει και τους δύο όρους της επερώτησης. Επίσης, η R_2 ευνοεί τα μικρά έγγραφα ($R_2(d_1, q) > R_2(d_4, q)$).

R_3

Λαμβάνει υπόψη όχι μόνο το ποσοστό του εγγράφου στο οποίο δεν έχουμε ταίριασμα αλλά και το ποσοστό της επερώτησης στο οποίο δεν έγινε ταίριασμα. Η συνάρτηση αυτή επιστρέφει 1 , αυστηρά μόνο στην περίπτωση που το έγγραφο είναι απόλυτα σχετικό με την επερώτηση, δηλαδή περιέχει μόνο τους όρους της επερώτησης.

R_4

Ένα έγγραφο που περιέχει πολλές φορές τους όρους της επερώτησης με βάση την R_3 θα έχει την ίδια συνάφεια με ένα έγγραφο που περιέχει μία μόνο φορά τους όρους της επερώτησης, ενώ με βάση την R_4 θα έχει μικρότερη συνάφεια. Η συμπεριφορά αυτή της R_3 είναι αποδεκτή καθώς τα έγγραφα θεωρούνται σύνολα όρων. Τα R_1, R_2, R_3 παίρνουν σαν μέγιστη τιμή το 1, ενώ η μέγιστη τιμή που μπορεί να πάρει η R_4 είναι το 0.5 στην περίπτωση που το έγγραφο είναι απόλυτα σχετικό με την επερώτηση.

R_5

Η R_5 λαμβάνει υπόψη μόνο τους κοινούς όρους της επερώτησης με το έγγραφο. Η συνάρτηση αυτή δεν είναι κανονικοποίημενη. Όσο περισσότεροι είναι οι κοινοί όροι της επερώτησης με το έγγραφο τόσο μεγαλύτερη τιμή λαμβάνει.

Το μοντέλο ανάκτησης βλέπει τα έγγραφα και τις επερωτήσεις σαν σύνολα όρων. Άρα, το σύνολο "a b c" θεωρείται το ίδιο με το σύνολο "a b c c" . Δηλαδή δεν μας ενδιαφέρει πόσες φορές εμφανίζεται ένας όρος. Οπότε, στο παράδειγμα μας θέλουμε τα d_1, d_2 να έχουν την ίδια διαβάθμιση, εφόσον θεωρούνται το ίδιο σαν σύνολα. Ομοίως, για τα d_3, d_4. Οι συναρτήσεις που διαβαθμίζουν το ίδιο τα d_1, d_2 είναι οι R_1, R_3, R_5. Οι συναρτήσεις που διαβαθμίζουν το ίδιο τα d_3, d_4 είναι οι R_1, R_3, R_5. Όμως, οι R_1, R_5 δίνουν τον ίδιο βαθμό συνάφειας στα d_1, d_2, d_3, d_4. Οπότε, μπορούμε να πούμε ότι η R_3 δίνει καλύτερα αποτελέσματα.

Άσκηση 3 (2 βαθμοί)
Θεωρείστε το ακόλουθο ευρετήριο το οποίο έχει τη μορφή: term: doc1: ⟨ position1, position2, . . . ⟩; doc2: ⟨ position1, position2, . . . ⟩; κλπ.

$$
\begin{aligned}
angels &: 2:\langle 36,174,252,651\rangle; 4:\langle 12,22,102,432\rangle; 7:\langle 17\rangle;\\
fools &: 2:\langle 1,17,74,222\rangle; 4:\langle 8,78,108,458\rangle; 7:\langle 3,13,23,193\rangle;\\
fear &: 2:\langle 87,704,722,901\rangle; 4:\langle 13,43,113,433\rangle; 7:\langle 18,328,528\rangle;\\
in &: 2:\langle 3,37,76,444,851\rangle; 4:\langle 10,20,110,470,500\rangle; 7:\langle 5,15,25,195\rangle;\\
rush &: 2:\langle 2,66,194,321,702\rangle; 4:\langle 9,69,149,429,569\rangle; 7:\langle 4,14,404\rangle;\\
to &: 2:\langle 47,86,234,999\rangle; 4:\langle 14,24,774,944\rangle; 7:\langle 199,319,599,709\rangle;\\
tread &: 2:\langle 57,94,333\rangle; 4:\langle 15,35,155\rangle; 7:\langle 20,320\rangle;\\
where &: 2:\langle 67,124,393,1001\rangle; 4:\langle 11,41,101,421,431\rangle; 7:\langle 16,36,736\rangle;
\end{aligned}
$$

Ποιά έγγραφα (αν υπάρχουν) ικανοποιούν τα επόμενα ερωτήματα·
q_1: "fools rush in"
q_2: "fools rush in" AND "angels fear to tread"
Τα εισαγωγικά σηματοδοτούν phrase queries.

ΛΥΣΗ

Ένα έγγραφο για να ικανοποιεί ένα phrase query θα πρέπει να υπάρχει στις λίστες όλων των όρων του query και οι θέσεις των όρων στα έγγραφα πρέπει να είναι διαδοχικές και να ακολουθούν τη σειρά εμφάνισης στην επερώτηση.

Για να βρούμε ποια έγγραφα ικανοποιούν το phrase query q1:"fools rush in" θα πρέπει αρχικά να ψάξουμε στο ευρετήριο για τα έγραφα που περιέχουν και τις τρεις λέξεις από τις οποίες αποτελείται. Αυτά είναι τα έγγραφα 2,4,7. Στη συνέχεια, πρέπει να ελέγξουμε αν τα positions των όρων είναι διαδοχικά.
Βλέπουμε ότι στα έγγραφα 2,4,7 ισχύουν τα εξής:

- *έγγραφο 2*: εμφανίζεται ο όρος "fools" στη θέση 1, ο όρος "rush" στη θέση 2 και ο όρος "in" στη θέση 3.
- *έγγραφο 4*: εμφανίζεται ο όρος "fools" στη θέση 8, ο όρος "rush" στη θέση 9 και ο όρος "in" στη θέση 10.
- *έγγραφο 7*: ο όρος "fools" βρίσκεται στις θέσεις 3 και 13, ο όρος "rush" βρίσκεται στις θέσεις 4 και 14, και ο όρος "in" βρίσκεται στις θέσεις 5 και 15.

Άρα, τα έγγραφα που ικανοποιούν το q1="fools rush in" είναι τα 2, 4 και 7.

Για την επερώτηση q2="fools rush in" AND "angels fear to tread" , για να βρούμε τα έγγραφα που την ικανοποιούν πρέπει να βρούμε τα εγγράφα που ικανοποιούν την υποερώτηση "fools rush in" , τα έγγραφα που ικανοποιούν την υποερώτηση "angels fear to tread" και να πάρουμε την τομή τους.
Τα έγγραφα που ικανοποιούν την υποερώτηση "fools rush in" τα έχουμε βρει προηγουμένως και είναι τα 2, 4 και 7.

Βλέπουμε ότι το *έγγραφο 4*, ικανοποιεί την υποερώτηση "angels fear to tread" γιατί σ΄ αυτό εμφανίζεται ο όρος "angels" στη θέση 12, ο όρος "fear" στη θέση 13, ο όρος "to" στη θέση 14, και ο όρος "tread" στη θέση 15.

Τέλος, βρίσκουμε την τομή των εγγράφων 2,4,7 και 4. Άρα, το έγγραφο 4 ικανοποιεί το q2="fools rush in" AND "angels fear to tread".

Άσκηση 4 (4 βαθμοί)

Θεωρείστε μια μηχανή αναζήτησης η οποία στηρίζεται σε ανεστραμμένο ευρετήριο και υποστηρίζει το Λογικό Μοντέλο (Boolean Model). Για κάθε λέξη το ευρετήριο μας δίνει μια μέθοδο που μας επιστέφει όλα τα αναγνωριστικά των εγγράφων που την περιέχουν σε αύξουσα σειρά. Θεωρείστε τις εξής επερωτήσεις:

q_1: μέλι AND ουρανός

q_2: μέλι AND ουρανός AND πορτοκάλι

q_3: (πορτοκάλι OR δέντρα) AND (μέλι OR ουρανός) AND (όραση OR μάτια)

Επίσης θεωρείστε τα εξής μεγέθη των λιστών εμφάνισης (posting list sizes):

Όρος	Μέγεθος Λίστας
δέντρα	316812
μάτια	213312
μέλι	107913
όραση	87009
ουρανός	271658
πορτοκάλι	46653

(α) Δώστε ένα γρήγορο αλγόριθμο για την αποτίμηση του q_1

(β) Δώστε ένα γρήγορο αλγόριθμο για την αποτίμηση του q_2 (εστιάστε στη σειρά με την οποία συμφέρει να γίνει η αποτίμηση)

(γ) Δώστε ένα γρήγορο αλγόριθμο για την αποτίμηση του q_3 (εστιάστε στη σειρά με την οποία συμφέρει να γίνει η αποτίμηση)

Σημείωση: Για τον υπολογισμό των παραπάνω μπορείτε να φτιάξετε σχετικό πρόγραμμα (χωρίς αυτό να είναι απαραίτητο).

ΛΥΣΗ

α)

Για να αποτιμήσουμε το query q_1="μέλι AND ουρανός " με βάση το ανεστραμμένο ευρετήριο και το Λογικό Μοντέλο ανάκτησης πρέπει να γίνουν τα εξής:

(1) Εντοπισμός του όρου μέλι στο ευρετήριο

(2) Ανάκτηση της posting list του

(3) Εντοπισμός του όρου ουρανός στο ευρετήριο

(4) Ανάκτηση της posting list του

(5) Εύρεση της τομής των δύο postings lists

Η διαδικασία της τομής (postings list intersection) είναι πολύ σημαντική. Μας επιτρέπει να βρίσκουμε γρήγορα τα έγγραφα που περιέχουν και τους δύο όρους. Η διαδικασία αυτή αναφέρεται και ως postings lists merge. Ο όρος αλγόριθμος συγχώνευσης (merge algorithm) χρησιμοποιείται για μια οικογένεια αλγορίθμων που συνδυάζουν ταξινομημένες λίστες. Εδώ κάνουμε συγχώνευση λιστών με τον λογικό τελεστή AND.

Ένας απλός και αποτελεσματικός τρόπος υπολογισμού της τομής των postings lists είναι ο εξής αλγόριθμος συγχώνευσης. Διατηρούμε δείκτες στις δύο λίστες και τις διασχίζουμε ταυτόχρονα, σε χρόνο γραμμικό σε σχέση με τον αριθμό των postings entries. Σε κάθε βήμα, συγκρίνουμε τα docID που δείχνουν οι δυο δείκτες. Εάν είναι οι ίδιοι, τότε βάζουμε το συγκεκριμένο docID στην λίστα με τα αποτελέσματα (results list) και αυξάνουμε τους δείκτες. Διαφορετικά, αυξάνουμε το δείκτη προς τον μικρότερο docID. Εάν τα μήκη των postings lists είναι x και y,τότε η τομή τους χρειάζεται O(x + y) βήματα. Τυπικά, η πολυπλοκότητα υπολογισμού του query είναι Θ(N), όπου N είναι ο αριθμός των εγγράφων στη συλλογή. Για να χρησιμοποιηθεί αυτός ο αλγόριθμος, είναι εξαιρετικά μεγάλης σημασίας οι λίστες να είναι ταξινομημένες με το ίδιο κριτήριο. Εδώ οι λίστες είναι ταξινομημένες με βάση το docID. Ο αλγόριθμος υπολογισμού της τομής των postings lists p1 και p2 μπορεί να γραφτεί ως εξής:

```
INTERSECT(p1, p2)
answer = NIL
while p1 <> NIL and p2 <> NIL
do if docID(p1) = docID(p2)
    then ADD(answer, docID(p1))
        p1 = next(p1)
        p2 = next(p2)
    else if docID(p1) < docID(p2)
        then p1 = next(p1)
        else p2 = next(p2)
return answer
```

β)

Η διαδικασία εύρεσης της τομής μπορεί να επεκταθεί και για πιο σύνθετες επερωτήσεις, όπως την q_2="μέλι AND ουρανός AND πορτοκάλι". Για κάθε ένα από τους t όρους, απαιτείται να πάρουμε τις postings lists τους, και να βρούμε την τομή τους. Η καθιερωμένη μέθοδος είναι η επεξεργασία των όρων με αύξουσα σειρά συχνότητας εμφάνισης στα έγγραφα(document frequency): εάν ξεκινήσουμε με την τομή των δύο μικρότερων postings lists, τότε όλα τα ενδιάμεσα αποτελέσματα πρέπει να μην είναι μεγαλύτερα από τη μικρότερη postings list και επομένως έτσι πιθανόν να κάνουμε την ελάχιστη δουλειά. Άρα, για τα postings lists των όρων "μέλι"(size=107913), "ουρανός"(size=271658), "πορτοκάλι"(size=46653),η επερώτηση q_2 εκτελείτε με την εξής σειρά: (πορτοκάλι AND μέλι) AND ουρανός

Ο αλγόριθμος για conjunctive queries,ο οποίος επιστρέφει το σύνολο των εγγράφων που περιέχουν κάθε όρο που δίνεται σαν παράμετρος, μπορεί να γραφτεί ως εξής:

```
INTERSECT(t1, . . . , tn)
terms =SortByIncreasingFrequency(t1, . . . , tn)
result=  postings( first(terms))
terms = rest(terms)
while terms <> NIL and result <> NIL do
    result = INTERSECT(result, postings( first(terms)))
    terms = rest(terms)
return result
```

γ)

Για πιο γενικές επερωτήσεις, όπως η q_3="(πορτοκάλι OR δέντρα) AND (μέλι OR ουρανός) AND (όραση OR μάτια)", μπορούμε να εργαστούμε όπως προηγουμένως. Βρίσκουμε τις συχνότητες όλων των όρων, και μετά υπολογίζουμε το μέγεθος κάθε OR,το οποίο είναι το άθροισμα των συχνοτήτων των όρων της διάζευξης. Τέλος, επεξεργαζόμαστε την επερώτηση σε αύξουσα σειρά μεγέθους των διαζευτικών όρων. Εδώ η διάζευξη "πορτοκάλι OR δέντρα" μπορεί να έχει μέγιστο μέγεθος 46653+316812=363465, η διάζευξη "μέλι OR ουρανός" μπορεί να έχει μέγιστο μέγεθος 107913+271658=379571 και η διάζευξη "όραση OR μάτια" μπορεί να έχει μέγιστο μέγεθος 87009+213312=300321 έγγραφα. Άρα, η σειρά με την οποία θα γίνουν οι συζεύξεις είναι η εξής: ((όραση OR μάτια) AND (πορτοκάλι OR δέντρα)) AND (μέλι OR ουρανός)