Γιάννης Μπάρζας 2765

**4η Εργασία στα σύνθετα δεδομένα**

**Μέρος 1:**

Διαβαζω το αρχειο και το αποθηκευω γραμμη γραμμη στη λιστα restaurant\_list.Στην συνεχεια διαβαζω την λιστα και δημιουργω το λεξικο inverted\_ ( το αντεστραμμένο αρχείο ) που εχει για κλειδι ταγκ και τιμη μια λιστα με τις θεσεις στην λιστα restaurant\_list των γραμμων του αρχειου που περιεχουν τα ταγκ .Μετα για να βρω τα εστιατορια που περιεχουν τα ταγκ που περνω από τον χρηστη περνω τις ταξονομημενες λιστες των ταγκ και κανω merge sort ανα 2 λιστες για να βρω την τομη τους και το αποτελεσμα με την επομενη(εχω και την περιπτωση ενός ταγκ).Στην raw αναζητηση απλα ξανα διαβαζω την λιστα που κραταω το αρχειο και περνω τις θεσεις οπου οι εγγραφες εχουν όλα τα ταγκ.Οι συχνοτητες υπολογιζονται μεσω του inverted\_ ( το αντεστραμμένο αρχείο ) οπου στην ουσια η συχνοτητα ενός ταγκ είναι το μεγεθως της λιστας του ταγκ.

**Μέρος 2:**

Διαβαζω το αρχειο και το αποθηκευω γραμμη γραμμη στη λιστα restaurant\_list και ταυτοχρονα υπολογιζω τα min x,y και max x,y ετσι ώστε να υπολογισω το βημα για τα x,y και να κατασκευασω τον πινακα 50X50 grid. Στην συνεχεια στις λιστες x\_borders,y\_borders αποθηκευω τα ορια για κάθε τιμη του x,y και μετα γεμιζω τον πινακα grid αναλογα σε ποια ορια πεφτει το κάθε εστιατοριο.Εδω εχω ξεχωριστη περιπτωση για τις τιμες οπου το x η το y είναι 49 γιατι λογω στρογγυλοποιησης μπορει να μην είναι ακριβως στις max τιμες.Ο αλγοριθμος αναζητησης με βαση το grid διαβαζει έναν ένα τα ορθογωνια κελια του grid και για αυτά που δεν είναι κενα και εχουν στην συνθηκη isIntersected True με βαση το παραθυρο της ερωτησης και το παραθυρο του κελιου (ορθογωνιο) ελενχουν ποια από τα εστιατορια του ορθογωνιου περιεχονται οντως μεσα στα ορια του range query window και τα κρατα στο αποτελεσμα.Στην raw διαβαζω την λιστα του αρχειου και κραταω τις γραμμες που εχουν συντεταγμενες μεσα στο στα ορια του range query window.

**Μέρος 3:**

Σε αυτό τα μερος συνδιαζω απλα τα ιδια ζειτουμενα από τα μερη 1,2 όπως ακριβως ζητειται.

**Μέρος 4:**

Α) Για το α είναι φανερο οτι η kwSearchIF είναι γρηγοροτερη σε χρόνο εκτελεσης από την kwSearchRaw γιατι δεν χρειαζεται να ψαξουμε σε ολο το αρχειο αλλά μονο μεσα στο λεξικο (ανεστραμμενο αρχείο). Ομως εχει το μειονεκτημα οτι απαιτει μνημη για το λεξικό και για τη λιστα του αρχείου.

Β) Και εδώ η spaSearchGrid είναι πιο γρηγορη σε χρόνο εκτέλεσης από την spaSearchRaw καθως τώρα εχουμε καταταξει τα εστιατορια σε ορθογωνιες περιοχες, άρα το ψάξιμο για την περιοχή είναι πιο γρηγορο αφου ξέρουμε ποια ορθογώνια του grid είναι μέσα σε αυτά το ορθογώνιο της ερώτησης. Επίσης και εδώ υπαρχει το μειονέκτημα στο ότι απαιτειται μνημη για το grid πίνακα και για το ίδιο το αρχειο.

Γ) Οπως είναι προφανες από τα ερωτήματα α και β ο kwSpaSearchRaw θα είναι ο πιο αργός από τους άλλους δύο αλγοριθμους αναζητησης επειδη δεν χρησιμοποιεί κάποιο ευρετήριο ή άλλη βοηθητική δομη.Αλλα παρατηρείται ότι ο kwSpaSearchIF είναι αρκετα πιο γρηγορος απο τον kwSpaSearchGrid. Αυτό συνβαινει γιατι η αναζητηση με χρηση των tags μειώνουν(φιλτραρουν) πολυ περισσοτερα πιθανα εστιατορια σε σχεση με με αυτά πο φιλτραρονται κατά την αναζητηση με τοποθεσια.