ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΥΕ041 - ΠΛΕ081: Διαχείριση Σύνθετων Δεδομένων

(ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2018-19)

ΕΡΓΑΣΙΑ 2 – Χωρικά Δεδομένα

Προθεσμία: 6 Μαΐου 2019, 9μ.μ.

Στόχος της εργασίας είναι η ανάπτυξη τεχνικών δεικτοδότησης (δηλ. ευρετηρίασης) και αναζήτησης χωρικών δεδομένων.

Μέρος 1 (30%, δημιουργία ευρετηρίου)

Κατεβάστε το αρχείο Beijing_restaurants.txt από το ecourse. Το αρχείο περιέχει τις συντεταγμένες 51970 **σημείων** τα οποία είναι θέσεις εστιατορίων στο Πεκίνο. Η πρώτη γραμμή του αρχείου περιέχει το πλήθος των σημείων. Από την επόμενη γραμμή και μετά, κάθε γραμμή έχει τις συντεταγμένες (x και y) ενός εστιατορίου. Ζητείται να γράψετε ένα πρόγραμμα το οποίο θα φτιάχνει ένα απλό χωρικό ευρετήριο βασισμένο σε σχάρα (grid). Η σχάρα χωρίζει το χώρο που καλύπτουν τα σημεία σε 10*10 = 100 ισομεγέθη ορθογώνια (**κελιά**).

Για να δημιουργήσετε το grid θα πρέπει να διαβάσετε τα σημεία από το αρχείο και να βρείτε τη μικρότερη και τη μεγαλύτερη τιμή σε κάθε συντεταγμένη (x και y). Σε καθένα από τα σημεία, δώστε ένα αναγνωριστικό αριθμί (identifier), που θα αντιστοιχεί στη γραμμή στην οποία βρίσκεται στο αρχείο Beijing_restaurants.txt. Π.χ. το σημείο (39.856138,116.42394) παίρνει identifier 1, το σημείο (39.813336,116.486149) παίρνει identifier 2, κλπ. Κατόπιν, χωρίστε το εύρος τιμών σε κάθε συντεταγμένη σε 10 ίσα διαστήματα τιμών. Μετά, ταξινομήστε τα σημεία με βάση το κελί στο οποίο βρίσκονται. Δηλαδή, όλα τα σημεία στο κελί (0,0), πρέπει να είναι πριν από όλα τα σημεία στο κελί (0,1), κλπ. Γράψτε τα ταξινομημένα σημεία σε ένα αρχείο grid.grd όπου κάθε γραμμή είναι της μορφής <identifier x-coordinate y-coordinate>. Παράλληλα δημιουργείστε και ένα άλλο αρχείο grid.dir (directory), το οποίο θα έχει στην πρώτη γραμμή του την ελάχιστη και τη μέγιστη τιμή σε κάθε άξονα. Στις επόμενες γραμμές, για κάθε κελί θα υπάρχει η συντεταγμένη του κελιού (π.χ. (0,0), (0,1) κλπ), η θέση στο αρχείο grid.grd από την οποία ξεκινά η γραμμή που περιέχει το πρώτο σημείο στο κελί, και ο αριθμός των σημείων στο κελί. Έτσι, γνωρίζοντας τις συντεταγμένες ενός κελιού, θα μπορεί κάποιος να χρησιμοποιεί τα δεδομένα του dir αρχείου για να εντοπίσει και να διαβάσει όλα τα σημεία που περιέχονται σε αυτό χωρίς να διαβάσει δεδομένα εκτός κελιού.

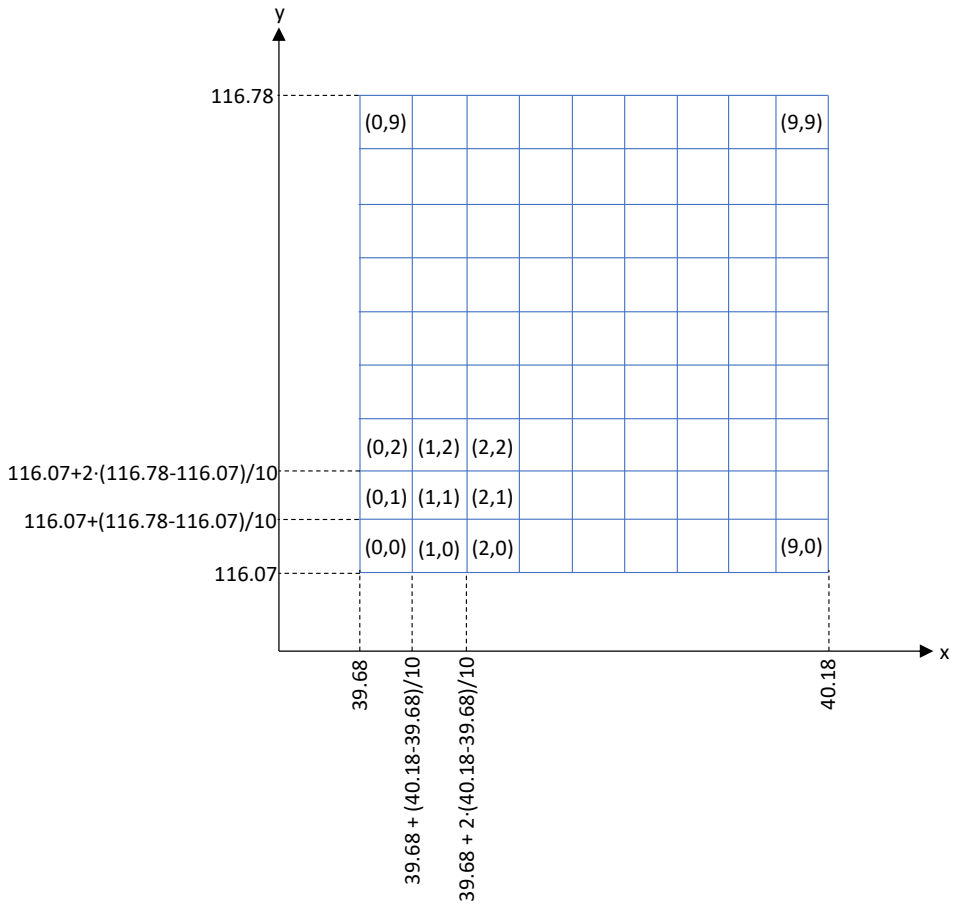
Παρακάτω δίνεται ένα παράδειγμα των δύο αρχείων καθώς και ένα σχηματικό παράδειγμα του grid. Στο παράδειγμα του grid δείχνονται οι συντεταγμένες των διαχωριστικών γραμμών των κελιών καθώς επίσης και οι συντεταγμένες (0,0), (0,1), κλπ. των κελιών στη 10*10 σχάρα. Κάθε σημείο αντιστοιχίζεται στο κελί στο οποίο περιέχεται. Αν ένα σημείο πέφτει ακριβώς σε μια διαχωριστή γραμμή, αντιστοιχίζεται στο κελί που ακολουθεί, π.χ. ένα σημείο με x συντεταγμένη ακριβώς 39.68 + (40.18-39.68)/10 και y 116.072 πέφτει στο κελί (1,0).

Πρώτες γραμμές του grid.dir

```
39.680090 40.179911 116.070466 116.719976
0 0 0  108
0 1 2894 179
0 2 7688 20
0 3 8226 117
0 4 11368 356
0 5 20915 91
0 6 23352 58
0 7 24907 18
1 0 25393 83
```

Πρώτες γραμμές του grid.grd

```
56 39.729270 116.119278
573 39.729398 116.128704
1253 39.723127 116.121828
1372 39.729585 116.127883
1395 39.729571 116.128738
2692 39.706018 116.123828
2846 39.727021 116.121720
3427 39.726623 116.120578
3804 39.723701 116.123683
4146 39.728675 116.135041
```

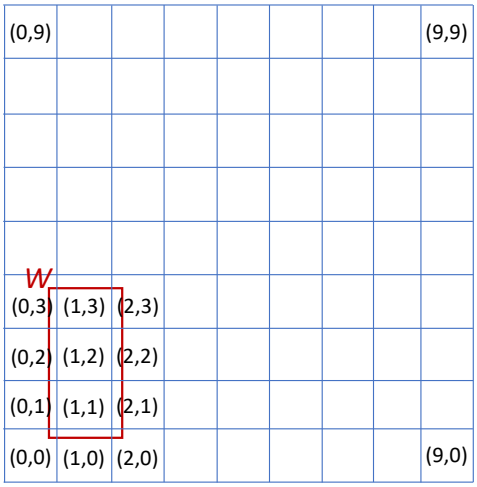


Ελέγξτε την ορθότητα του προγράμματός σας, διασταυρώνοντας τα περιεχόμενα του grd και dir αρχείου με αυτά του Beijing_restaurants.txt.

Μέρος 2 (30%, ερωτήματα επιλογής)

Ζητείται να γράψετε ένα πρόγραμμα το οποίο θα χρησιμοποιεί το grid που φτιάξατε στο 1ο μέρος ώστε να αποτιμήσετε ερωτήματα επιλογής παραθύρου (window selection queries). Το πρόγραμμα θα παίρνει σαν command-line arguments το κάτω και το πάνω όριο του παραθύρου σε κάθε διάσταση, δηλ. τις τιμές <x_low> <x_high> <y_low> <y_high>. Θα πρέπει να υπολογίζει και να τυπώνει στην έξοδο τα σημεία που περιλαμβάνονται στο παράθυρο.

Κατ' αρχήν το πρόγραμμα θα διαβάζει εξ ολοκλήρου το αρχείο grid.dir και θα αποθηκεύει σε μια δομή στη μνήμη τα περιεχόμενά του, τα οποία θα χρησιμοποιήσει για την αποτίμηση (evaluation). Κατόπιν θα διαβάζει τα δεδομένα των σημείων για κάθε κελί που επικαλύπτεται με το παράθυρο. Αν το κελί καλύπτεται πλήρως από το παράθυρο τα σημεία του κελιού θα βγαίνουν στην έξοδο. Αν το κελί καλύπτεται μερικώς από το παράθυρο, τότε θα πρέπει να επαληθεύσουμε για κάθε σημείο στο κελί αν το σημείο είναι όντως μέσα στο παράθυρο. Στο παράδειγμα ερωτήματος *W* που εμφανίζεται στην παρακάτω εικόνα, για τα κελιά (1,2) και (1,1) δίνουμε στην έξοδο όλα τα σημεία τους χωρίς να εξετάζουμε τις συντεταγμένες τους, ενώ για τα κελιά (0,3),(1,3),(2,3),(0,2),(2,2),(0,1),(2,1),(0,0),(1,0) και (2,0) πρέπει να συγκριθούν οι συντεταγμένες των σημείων με αυτές του παραθύρου *W* ώστε να επιβεβαιωθεί το αν περικλείονται σε αυτό.



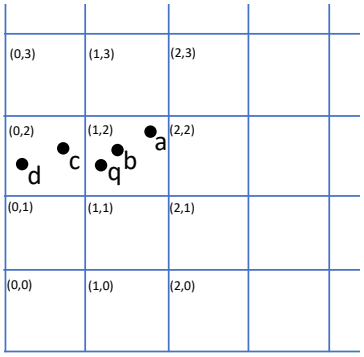
Κάντε έλεγχο ορθότητας, συγκρίνοντας το αποτέλεσμα με αυτό που θα παίρνατε αν αποτιμούσατε το ερώτημα απ' ευθείας στο αρχείο Beijing_restaurants.txt. Επίσης, για τον έλεγχο, να μετράτε και να επιστρέφετε τον αριθμό των κελιών των οποίων τα περιεχόμενα διαβάστηκαν σε κάθε ερώτημα.

Μέρος 3 (40%, ερωτήματα πλησιέστερου γείτονα)

Ζητείται να γράψετε ένα πρόγραμμα το οποίο θα χρησιμοποιεί το grid που φτιάξατε στο 1ο μέρος ώστε να αποτιμήσετε ερωτήματα πλησιέστερου γείτονα σταδιακά (incremental nearest neighbor search). Το πρόγραμμα θα παίρνει σαν command-line arguments έναν αριθμό k και τις δύο συντεταγμένες ενός σημείου q του οποίου θα πρέπει να τυπώνει τα k πλησιέστερα εστιατόρια. Αρχικά το πρόγραμμα θα διαβάζει εξ ολοκλήρου το αρχείο grid.dir και θα αποθηκεύει σε μια δομή στη μνήμη τα περιεχόμενά του, τα οποία θα χρησιμοποιήσει για την αποτίμηση (evaluation). Η συνάρτηση εύρεσης των πλησιέστερων γειτόνων πρέπει να υλοποιηθεί σαν iterator (π.χ. generator function) έτσι ώστε σε κάθε κλήση της να επιστρέφει το επόμενο πλησιέστερο σημείο. Το πρόγραμμα θα πρέπει να καλέσει τη συνάρτηση k φορές για να επιστρέψει και να τυπώσει τα k κοντινότερα στο q σημεία.

Η συνάρτηση θα πρέπει να διαχειρίζεται μια δομή (ουρά προτεραιότητας) στην οποία μπαίνουν κελιά και σημεία τα οποία οργανώνονται με βάση την απόστασή τους από το q. Στην αρχικοποίησή της η συνάρτηση βάζει μέσα στη δομή μόνο το κοντινότερο κελί στο q (π.χ. αυτό που το περιέχει). Η συνάρτηση κάθε φορά βρίσκει το κοντινότερο στοιχείο στο q (δηλ. το στοιχείο στην κορυφή της ουράς) και το αφαιρεί από τη δομή. Αν το στοιχείο αυτό είναι κελί, τότε διαβάζει τα περιεχόμενα του κελιού (που είναι σημεία) από το grid.grd (εντοπίζοντάς τα μέσω της πληροφορίας από το grid.dir) και τα προσθέτει στη δομή. Επιπλέον, τα γειτονικά κελιά στο κελί που μόλις βγήκε από τη δομή, **τα οποία δεν έχουν ξαναμπεί στη δομή**, προστίθενται σε αυτή. Αν το στοιχείο που μόλις βγήκε από τη δομή είναι σημείο, τότε αυτό επιστρέφεται σαν το επόμενο πλησιέστερο στο q.

Στο παρακάτω παράδειγμα, το σημείο q βρίσκεται μέσα στο κελί (1,2) οπότε και αυτό το κελί μπαίνει πρώτο στη δομή. Κατόπιν, το κελί (1,2) βγαίνει και στη δομή μπαίνουν (i) τα σημεία a, b που βρίσκονται μέσα στο (1,2) και (ii) τα κελιά (0,3), (1,3), (2,3), (0,2), (2,2), (0,1), (1,1), (2,1) που είναι γειτονικά του (1,2). Το επόμενο πλησιέστερο στο q στοιχείο της δομής είναι το κελί (0,2), το οποίο βγαίνει και μπαίνουν στη θέση του τα σημεία c και d, ενώ δεν μπαίνει κανένα γειτονικό κελί του (0,2), επειδή τα έχουμε προσθέσει όλα στο παρελθόν. Το επόμενο στοιχείο που βγαίνει από τη δομή είναι το b. Είναι σημείο και άρα επιστρέφεται ως ο 1ος πλησιέστερος γείτονας. Στην επόμενη κλήση της συνάρτησης, αυτή συνεχίζει, προσθέτοντας τα περιεχόμενα του επόμενου πλησιέστερου κελιού (1,1), καθώς και τους γείτονες (0,0), (1,0), (2,0), μετά τα περιεχόμενα του (0,1). Μετά το επόμενο κοντινότερο στοιχείο είναι το σημείο c το οποίο επιστρέφεται ως ο επόμενος (δηλ. ο δεύτερος) πλησιέστερος γείτονας του q. Η συνάρτηση συνεχίζει με τον ίδιο τρόπο στις επόμενες κλήσεις.



Θα χρειαστεί να φτιάξετε μια συνάρτηση mindist η οποία υπολογίζει την (ελάχιστη) Ευκλείδεια απόσταση μεταξύ ενός σημείου (του q) και ενός κελιού. Η απόσταση μπορεί να συντεθεί από τις αποστάσεις σε κάθε άξονα όπως έχουμε πει στο μάθημα. Δηλαδή είναι η ρίζα του αθροίσματος των τετραγωνισμένων αποστάσεων σε κάθε διάσταση.

Βάλτε το πρόγραμμά σας να τυπώνει εκτός από τα k κοντινότερα στο q σημεία και τα κελιά τα οποία διαβάστηκαν από το πρόγραμμα.

Παραδοτέα: Κάντε turnin στο assignment2@mye041 τα προγράμματά σας και ένα PDF αρχείο το οποίο τα τεκμηριώνει και περιέχει τυχόν οδηγίες. Προσοχή: τα προγράμματά σας πρέπει να τρέχουν στα μηχανήματα του εργαστηρίου ΠΕΠ2, όπου και θα εξεταστείτε.