Διαχείριση Σύνθετων Δεδομένων
2η Σειρά Ασκήσεων

Πανεπιστήμιο Ιωαννίνων

Εαρινό 2020

Πασόη Σοφία ΑΜ:2798
Υπεύθυνος Καθηγητής: κ. Μαμουλής Νικόλαος

Μέρος 1: Κατασκευή R-δέντρου

1. **Η συνάρτηση readFile():**
   Το αρχείο data_rectangles.txt δίνεται ως όρισμα στην γραμμή εντολών. Παίρνει το αρχείο, το ανοίγει και ξεκινά το διάβασμα. Κάνει split στο tab για να πάρει κάθε αριθμό ξεχωριστά. Στη θέση του newMbr[0] βρίσκεται το object-id ενώ στις θέσεις 1,2,3 και 4 βρίσκονται οι πραγματαικοί x-low, x-high, y-low, y-high αντίστοιχα. Κάθε newMbr πίνακα που φτιάχνει τον βάζει σε έναν άλλο πίνακα(mbr) και τον επιστρέφει. Δξλαδή κάθε θέση του mbr είναι ένας πίνακας newMbr.

2. **Η συνάρτηση sortMbr(mbr):**
   Αρχικά ορίζει τις μεταβλητές r ως το μήκος του πίνακα, n ως των αριθμό των εγγραφών που χωράει ένας κόμβος, P ως το ανώφλι του r/n και S το ανώφλι του √P. Έπειτα ταξινομεί τα ορθογώνια με βάση το x-low. Στη συνέχεια κατασκευάζει τις S λίστες και τις ταξινομεί με βάση το y-low. Επειδή η τελευταία S λίστα μπορεί να περιέχει λιγότερα από S*n στοιχεία ταξινομείται έξω απο το for-loop, για όσα στοιχεία την αποτελούν.

3. **Η συνάρτηση splitMbr(mbr):**
   Παίρνει ως όρισμα τον mbr και ορίζει τις μεταβλητές r ως το μήκος του πίνακα, n ως των αριθμό των εγγραφών που χωράει ένας κόμβος και P ως το ανώφλι του r/n. Κατασκευάζει τους κόμβους βάζοντας τον καταλλήλο αριθμό εγγραφών και τους προσθέτει σε μία καινούρια λίστα newNodes. Επειδή η τελευταία λίστα μπορεί να μην γεμίσει η διαδικασία για αυτή γίνεται έξω από το for-loop, για όσα στοιχεία την αποτελούν.

4. **Η συνάρτηση addNodeIds(newNodes, n):**
   Παίρνει ως όρισμα τη λίστα newNodes με τους κόμβους και τη χωρητικότηκα κάθε κόμβου n. Δίνει σε κάθε εγγραφή του κόμβου έναν ακέραιο αριθμό id που προσδιορίζει τον αριθμό του κόμβου(π.χ. ο πρώτος κόμβος έχει αναγνωριστικό 0, ο δεύτερος 1 κτλ).

5. **Η συνάρτηση convertToMbr(nodes):**
   Παίρνει ως όρισμα τη λίστα με τους κόμβους nodes. Παίρνει κάθε εγγραφή του κόμβου και βρίσκει τις διαστάσεις και το αναγνωριστικό id του κάθε ορθογωνίου. Τα βάζει σε ένα πίνακα, τον οποίο και επιστρέφει.

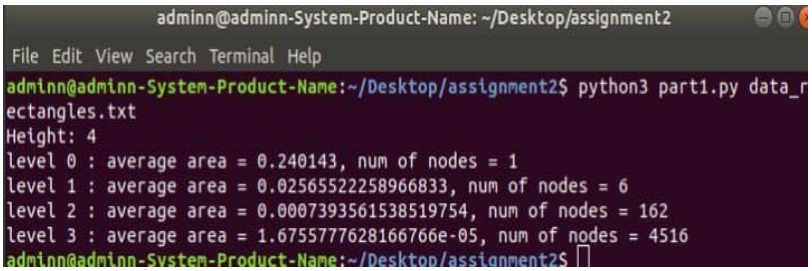
6. **Η συνάρτηση findAvgArea(mbr):**
   Παίρνει ως όρισμα τη λίστα με τα ορθογώνια. Για κάθε ορθογώνιο παίρνει τις τιμές x-low, x-high, y-low και y-high. Υπολογίζει το μήκος κάθε πλευράς(βάσης και ύψους). Το μήκος της βάσης υπολογίζεται ως x-high - x-low και αντίστοιχα του ύψους y-high - y-low. Από τον τύπο βάση επί ύψος υπολογίζεται το εμβαδόν του ορθογωνίου. Βρίσκει το άθροισμα όλων των εμβαδών των ορθογωνίων και το διαιρεί με το πλήθος τους ώστε να βρει το μέσο εμβαδόν. Τέλος, το επιστρέφει.

7. **Η συνάρτηση main():**
   Καλείται η συνάρτηση readFile() και το αποτέλεσμα της αποθηκεύεται σε μια μεταβλητή mbr. Όσο υπάρχουν γραμμές στο αρχείο καλούνται οι συναρτήσεις sortMbr και split με όρισμα τη μεταβλητή mbr. Το αποτέλεσμα της συνάρτησης split αποθηκεύεται σε μία μεταβλητή newMbr. Έπειτα καλέιται η addNodesIds με ορίσματα το newMbr και το μήκος της λίστας των κόμβων nodes. Η λίστα nodes γίνεται extend με το newMbr. Στη συνέχεια αποθηκεύει το αποτέλεσμα της συνάρτησης convertToMbr με όρισμα newMbr στη μεταβλητή mbr. Με την νέα τιμή της μεταβλητής mbr καλείται η findAvgArea και το αποτελέσμα της αποθηκεύεται σε μια μεταβλητή a. Σε κάθε θέση του πίνακα info τοποθετείται η μεταβλητή a μαζί με το μέγεθος του (νέου)mbr. Άρα με την ολοκλήρωση του

while-loop ο πίνακας info θα αποτελείται από τόσες θέσεις όσες και ο αριθμός των επιπέδων. Σε κάθε θέση θα υπάρχει το μέσο εμβδαδόν των ορθογωνίων κάθε επιπέδου και ο αριθμός των κόμβων. Άρα για να τυπώσει τα σταστικά του δέντρου:

- Για το ύψος(θεωρούμε ότι είναι ο αριθμός των επιπέδων) συνεπώς τυπώνουμε το μέγεθος του πίνακα info.
- Για κάθε επίπεδο:
  I. Ο αριθμός του επιπέδου είναι ο αριθμός της θέσης του πίνακα info στην οποία βρίσκεται εκείνη τη στιγμή.
  II. Το μέσο εμβαδόν βρίσκεται στη θέση 0 της κάθε θέσης του πίνακα info.
  III. Ο αριθμός των κόμβων βρίσκεται στη θέση 1 της κάθε θέσης του πίνακα info.

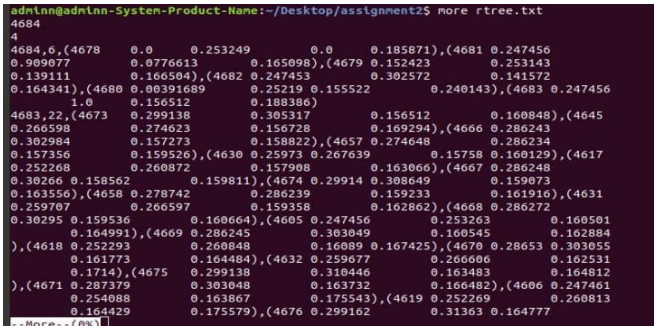
Επειδή το επίπεδο 0 αναφέρεται στη ρίζα και επειδή στη θέση 0 του πίνακα info βρίσκονται οι πληροφορίες για τα φύλλα και στην τελευταία θέση οι πληροφορίες για τη ρίζα, εδώ όταν το i στο for-loop θα βρίσκεται στο 0 θα παίρνει τις πληροφορίες από το τελευταίο στοιχείο. Αντίστοιχα, όταν το i θα βρίσκεται στη θέση 1 θα παίρνει τις πληροφορίες από το προτελευταίο στοιχείο και παέι λέγοντας. Συνπώς στην οθόνη θα τυπωθεί:



Για να γράψουμε τα ζητούμενα δεδομένα στο αρχείο, ανοίγουμε ένα αρχείο με όνομα rtree.txt και δικαιώματα write. Η πρώτη γραμμή θα έχει το αναγνωριστικό κόμβου id της ρίζας. το οποίο βρίσκεται στη θέση 0 της τελευταίας γραμμής της λίστας nodes. Η δεύτερη γραμμή θα περιέχει τον αριθμό των επιπέδων του δέντρου, ο οποίος είναι το μέγεθος του πίνακα info. Έπειτα επαναληπτικά(δαιτρέχω από το τέλος προς την αρχή για να γραφτούν με τη σωστή σειρά) για κάθε κόμβο θα τυπώνει το αναγνωριστικό id του κόμβου, των αριθμό των εγγραφών του κόμβου και τις αντίστοιχες πληροφορίες της κάθε εγγραφής. Πιο συγκεκριμένα στη λίστα nodes:

- Το αναγνωριστικό κόμβου id βρίσκεται στην πρώτη θέση κάθε γραμμής.
- Ο αριθμός των εγγραφών είναι το μέγεθος της γραμμής-1(-1 γιατί στην πρώτη θέση βρίσκεται το id).
- Οι πληροφορίες για κάθε εγγραφή βρίσκονται σε κάθε θέση της γραμμής(μετά το πρώτο στοιχείο που είναι το id), δηλαδή:
  1. nodes[i][j][0] είναι το ptr/object id.
  2. nodes[i][j][1] είναι το x-low.
  3. nodes[i][j][2] είναι το x-high.
  4. nodes[i][j][3] είναι το y-low.
  5. nodes[i][j][4] είναι το y-high.

Δείγμα αρχείου εξόδου:



Μέρος 2: Ερωτήσεις στο R-δέντρο

1. Συνάρτηση readRtree(filename):

Διαβάζει το αρχείο rtree.txt. Στην πρώτη γραμμή ξέρουμε πως βρίσκεται η ρίζα και στη δεύτερη ο αριθμός των επιπέδων του δέντρου. Συνεπώς αποθηκεύει τις τιμές τους σε δύο μεταβλητές root και levels αντίστοιχα. Επαναληπτικά για όλες τις γραμμές:

- Κάνει split στο "," .
- Παίρνει το πρώτο και το δεύτερο στοιχείο που είναι το αναγνωριστικό κόμβου id και ο αριθμός εγγραφών και τα βάζει σε ένα καινούριο πίνακα mbr στις θέσεις 0 και 1 αντίστοιχα.
- Επαναληπτικά για κάθε εγγραφή:
  - Κάνει split στο tab.
  - Το πρώτο στοιχείο είναι είναι ακέραιος(ptr/object id).
  - Τα υπόλοιπα 4 είναι οι πραγματικοί αριθμοί x-low, x-high, y-low και y-high.

  Δηλαδή για κάθε εγγραφή προκύπτει και ένας πίνακας toks, τον οποίο κάθε φορά τον βάζει στον πίνακα mbr. Κάθε θέση του mbr είναι της μορφής: (αναγνωριστικό κόμβου, αριθμός εγγραφών, [ptr/object id,x-low,x-high,y-low,y-high],[],....).
- Κάθε mbr που φτιάχνει το βάζει στη θέση 0 της λίστας rtree, έτσι ώστε η rtree που θα φτιαχτεί να ξεκινά από τη ρίζα και να πηγαίνει προς τα φύλλα.

Τέλος επιστρέφει τη λίστα rtree καθώς και τις μεταβλητές root και levels.

2. Συνάρτηση readFile(filename):

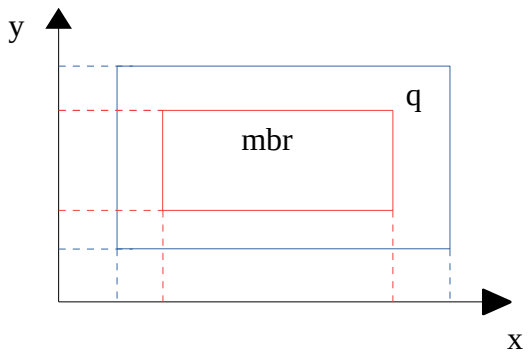
Παίρνει ως όρισμα ένα αρχείο. Το διαβάζει και επαναληπτικά για όλες τις γραμμές:

1. Κάνει split στο tab.
2. Το πρώτο στοιχείο είναι είναι ακέραιος(ptr/object id).
3. Τα υπόλοιπα 4 είναι οι πραγματικοί αριθμοί x-low, x-high, y-low και y-high.

Κάθε πίνακας newMbr που προκύπτει μπαίνει σε κάθε θέση ενός πίνακα mbr. Τέλος, ο πίνακας mbr επιστρέφεται.

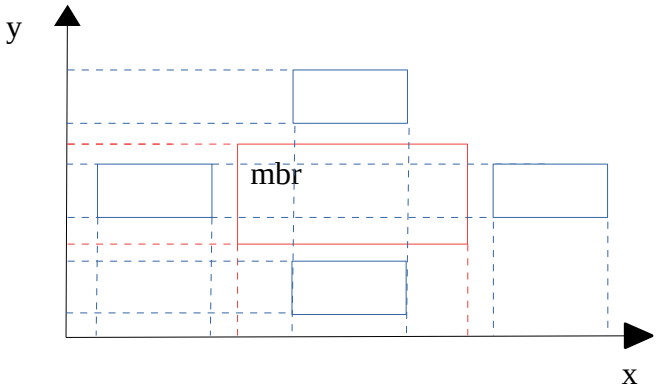
3. Συνάρτηση contains(q, mbr):

Παίρνει ως όρισμα 2 ορθογώνια και ελέγχει αν το q περιέχει το mbr. Πιο συγκριμένα αν το x-low του q είναι μικρότερο του x-low του mbr, το x-high του q είναι μεγαλύτερο του x-high του mbr, το y-low του q είναι μικρότερο του y-low του mbr και το y-high του q είναι μεγαλύτερο του y-high του mbr, τότε το το q περιέχει το mbr. Δηλαδή είναι στη μορφή:



4. Η συνάρτηση intersects(q, mbr):

Παίρνει ως όρισμα 2 ορθογώνια και ελέγχει αν το q τέμνει το mbr. Πιο συγκριμένα αν το x-high του q είναι μικρότερο του x-low του mbr και το x-low του q είναι μεγαλύτερο του x-high του mbr και το y-high του q είναι μικρότερο του y-low του mbr και το y-low του q είναι μεγαλύτερο του y-high του mbr, τότε το το q δεν τέμνει το mbr. Σε οποιαδήποτε άλλη περίπτωση το τέμνει. Δηλαδή δεν το τέμνει αν είναι σε κάποια από τις μορφές:



5. Η συνάρτηση intersectionQuery(mbr,q):

Μετράει πόσα ορθογώνια mbr τέμνουν το ορθογώνιο q. Για κάθε mbr θα καλείται η συνάρτηση intersects(q, mbr). Αν επιστρέφει true, σημαίνει ότι υπάρχουν κοινά σημεία και τότε θα αυξάνει το μετρητή κατά ένα. Αν όχι κρατάει την τιμή του μετρητή ως έχει. Τέλος, επιστρέφει το μετρητή.

6. Η συνάρτηση insideQuery(mbr,q):

Μετράει πόσα ορθογώνια mbr περιέχονται στο ορθογώνιο q. Για κάθε mbr θα καλείται η συνάρτηση contains(q, mbr). Αν επιστρέφει true, σημαίνει ότι το q περιέχει το mbr και τότε θα αυξάνει το μετρητή κατά ένα. Αν όχι κρατάει την τιμή του μετρητή ως έχει. Τέλος, επιστρέφει το μετρητή.

7. Η συνάρτηση containmentQuery(q,mbr):

Μετράει πόσα ορθογώνια mbr περιέχουν το ορθογώνιο q. Για κάθε mbr θα καλείται η συνάρτηση contains(mbr,q). Αν επιστρέφει true, σημαίνει ότι το q περιέχεται στο mbr και τότε θα αυξάνει το μετρητή κατά ένα. Αν όχι κρατάει την τιμή του μετρητή ως έχει. Τέλος, επιστρέφει το μετρητή.

8. Η συνάρτηση intersectionQueryRTree(rtree,n,q,currentLevel,treeLevel):

Παίρνει ως ορίσματα το rtree, τη ρίζα, το ορθογώνιο q, το επίπεδο στο οποίο βρίσκεται και τα επίπεδα του δέντρου. Υπολογίζει πόσα ορθογώνια του δέντρου τέμνουν το ορθογώνιο q. Κρατάει μια global μεταβλητή mylist και έναν μετρητή. Βάζει το περιεχόμενο της ρίζας του δέντρου στο mylist. Στη συνέχεια ελέγχει αν βρίσκεται στα φύλλα, ελέγχοντας αν το επίπεδο στο οποίο βρίσκεται είναι επίπεδα του δεντρου-1(-1 γιατί η αρίθμηση ξεκινά από το 0). Αν ισχύει η συνθήκη, τότε για κάθε ορθογώνιο καλέιται η intersects(q,tree[][]). Αν επιστρέφει true, σημαίνει ότι το q τέμνει το ορθογώνιο και τότε θα αυξάνει το μετρητή κατά ένα. Αν όχι κρατάει την τιμή του μετρητή ως έχει. Τέλος, επιστρέφει το μετρητή. Αν η συνθήκη δεν ισχύει, δηλαδή δε βρίσκεται σε κάποιο φύλλο, τότε για κάθε ορθογώνιο καλέιται η intersects(q,tree[][]). Αν επιστρέφει true, σημαίνει ότι το q τέμνει το ορθογώνιο και τότε παίρνει το αναγνωριστικό του κόμβου id. Η τιμή του μετρητή αυξάνεται σύμφωνα με την τιμή επιστροφής της αναδρομικής κλήσης της intersectionQueryRTree. Η αναδρομική κλήση παίρνει τα ίδια ορίσματα, εκτός από το n(ρίζα), που τη θέση του πήρε αναγνωριστικό του κόμβου id. Αν όχι κρατάει την τιμή του μετρητή ως έχει. Τέλος,

επιστρέφει το μετρητή(ξεκινά με αναγνωριστικό κόμβου αυτό της ρίζας και μετά το n παίρνει την τιμή του αναγνωριστικού κόμβου στον οποίο βρισκόμαστε).

9. Η συνάρτηση insideQueryRTree(rtree,n,q,currentLevel,treeLevel):

Παίρνει ως ορίσματα το rtree, τη ρίζα, το ορθογώνιο q, το επίπεδο στο οποίο βρίσκεται και τα επίπεδα του δέντρου. Υπολογίζει πόσα ορθογώνια του δέντρου περιέχονται στο ορθογώνιο q. Κρατάει μια global μεταβλητή mylist και έναν μετρητή. Βάζει το περιεχόμενο της ρίζας του δέντρου στο mylist. Στη συνέχεια ελέγχει αν βρίσκεται στα φύλλα, ελέγχοντας αν το επίπεδο στο οποίο βρίσκεται είναι επίπεδα του δεντρου-1(-1 γιατί η αρίθμηση ξεκινά από το 0). Αν ισχύει η συνθήκη, τότε για κάθε ορθογώνιο καλέιται η contains(q,tree[][]). Αν επιστρέφει true, σημαίνει ότι το q περιέχει το ορθογώνιο και τότε θα αυξάνει το μετρητή κατά ένα. Αν όχι κρατάει την τιμή του μετρητή ως έχει. Τέλος, επιστρέφει το μετρητή. Αν η συνθήκη δεν ισχύει, δηλαδή δε βρίσκεται σε κάποιο φύλλο, τότε για κάθε ορθογώνιο καλέιται η contains(q,tree[][]). Αν επιστρέφει true, σημαίνει ότι το q περιέχειτο ορθογώνιο και τότε παίρνει το αναγνωριστικό του κόμβου id. Η τιμή του μετρητή αυξάνεται σύμφωνα με την τιμή επιστροφής της αναδρομικής κλήσης της insideQueryRTree. Η αναδρομική κλήση παίρνει τα ίδια ορίσματα, εκτός από το n(ρίζα), που τη θέση του πήρε αναγνωριστικό του κόμβου id. Αν όχι κρατάει την τιμή του μετρητή ως έχει. Τέλος, επιστρέφει το μετρητή(ξεκινά με αναγνωριστικό κόμβου αυτό της ρίζας και μετά το n παίρνει την τιμή του αναγνωριστικού κόμβου στον οποίο βρισκόμαστε).

10. Η συνάρτηση containmentQueryRTree(rtree,n,q,currentLevel,treeLevel):

Παίρνει ως ορίσματα το rtree, τη ρίζα, το ορθογώνιο q, το επίπεδο στο οποίο βρίσκεται και τα επίπεδα του δέντρου. Υπολογίζει σε πόσα ορθογώνια του δέντρου περιέχεται το ορθογώνιο q. Κρατάει μια global μεταβλητή mylist και έναν μετρητή. Βάζει το περιεχόμενο της ρίζας του δέντρου στο mylist. Στη συνέχεια ελέγχει αν βρίσκεται στα φύλλα, ελέγχοντας αν το επίπεδο στο οποίο βρίσκεται είναι επίπεδα του δεντρου-1(-1 γιατί η αρίθμηση ξεκινά από το 0). Αν ισχύει η συνθήκη, τότε βάζει στο mylist το αναγνωριστικό του κόμβου id και για κάθε ορθογώνιο καλέιται η contains(tree[][],q). Αν επιστρέφει true, σημαίνει ότι το q περιέχεται το ορθογώνιο και τότε θα αυξάνει το μετρητή κατά ένα. Αν όχι κρατάει την τιμή του μετρητή ως έχει. Τέλος, επιστρέφει το μετρητή. Αν η συνθήκη δεν ισχύει, δηλαδή δε βρίσκεται σε κάποιο φύλλο, τότε για κάθε ορθογώνιο καλέιται η contains(tree[][],q). Αν επιστρέφει true, σημαίνει ότι το q περιέχεται στο ορθογώνιο και τότε παίρνει το αναγνωριστικό του κόμβου id. Η τιμή του μετρητή αυξάνεται σύμφωνα με την τιμή επιστροφής της αναδρομικής κλήσης της containmentQueryRTree. Η αναδρομική κλήση παίρνει τα ίδια ορίσματα, εκτός από το n(ρίζα), που τη θέση του πήρε αναγνωριστικό του κόμβου id. Αν όχι κρατάει την τιμή του μετρητή ως έχει. Τέλος, επιστρέφει το μετρητή(ξεκινά με αναγνωριστικό κόμβου αυτό της ρίζας και μετά το n παίρνει την τιμή του αναγνωριστικού κόμβου στον οποίο βρισκόμαστε).

11. Η συνάρτηση main():

Διαβάζει τα αρχεία data_rectangles.txt και query_rectangles.txt και τα βάζει σε 2 μεταβλητές mbr και queries αντίστοιχα. Έπειτα, επαναληπτικά για κάθε query καλούνται οι συναρτήσεις intersectionQuery(mbr, queries[i]), insideQuery(mbr, queries[i]) και containmentQuery(mbr, queries[i]), των οποίων οι αποτιμήσεις αποθηκεύονται σε μετβλητές, οι οποίες τυπώνονται(είναι ο αριθμός των αποτελεσμάτων της ερώτησης). Έπειτα, καλεί την readRTree() και αποθηκεύει της αποτιμήσεις της σε 3 μεταβλητές, rtree, root και levels. Στη συνέχεια , επαναληπτικά για κάθε query καλούνται οι συναρτήσεις intersectionQueryRTree(rtree,root,queries[i],0,levels),

insideQueryRTree(rtree,root,queries[i],0,levels), containmentQueryRTree(rtree,root,queries[i],0,levels) και οι τιμές τους αποθηκεύονται σε μεταβλητές οι οποίες αντιπροσωπεύουν τον αριθμό αποτελεσμάτων των ερωτήσεων στο rtree. Μετά την κλήση κάθε συνάρτησης υπολογίζεται το μήκος της global λίστας mylist που αντιπροσωπεύει τον αριθμό των κόμβων που επισκέφθηκε. Αφού υπολογιστεί ο αριθμός αυτός η λίστα αδειάζει ώστε να υπολογιστεί ο αριθμός και για τις επόμενες συναρτήσεις. Τέλος τυπώνει τον αριθμό αποτελεσμάτων για κάθε ερωτηση καθώς και τον αριθμό των κόμβων που επισκέφθηκε για την αποτίμηση της.

Παρατηρήσεις:

- Τρέχουμε τα αρχεία πληκτρολογώντας στο τερματικό την εντολή:
  1ο : python3 part1.py data_rectangles.txt
  2ο : python3 part2.py data_rectangles.txt query_rectangles.txt rtree.txt
- Μπορούμε να δούμε το περιχόμενο των αρχείων στο τερματικό πληκτρολογώντας την εντολή: more "filename".
- Κάθε αρχείο αντιστοιχεί και σε ένα ερώτημα.