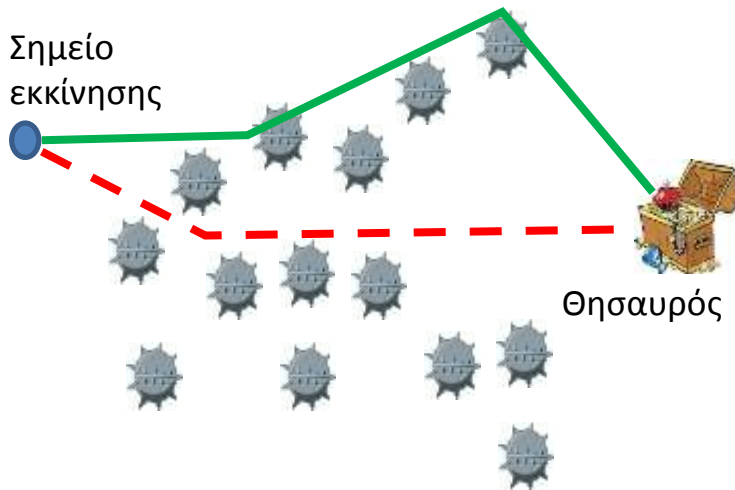


ΑΛΓΟΡΙΘΜΟΙ 2015-16 – ΥΠΟΧΡΕΩΤΙΚΗ ΑΤΟΜΙΚΗ ΕΡΓΑΣΙΑ α' σκέλος

Θέμα:

Σε μία τοποθεσία υπάρχει ένας θησαυρός με διαμάντια. Για να φτάσει κάποιος στο θησαυρό, θα πρέπει να αποφύγει μία περιοχή με νάρκες. Θέλουμε να βρούμε από το σημείο εκκίνησης, το συντομότερο μονοπάτι που να μην περνάει μέσα από το ναρκοπέδιο. Επιτρέπεται όμως να περνάει ξυστά από το ναρκοπέδιο, δηλαδή να περνάει πρακτικά πάνω από τις νάρκες περιμετρικά του ναρκοπεδίου.



Στο διπλανό σχήμα, το πράσινο μονοπάτι είναι έγκυρο, ενώ το κόκκινο δεν είναι.

Το σημείο εκκίνησης είναι πιο αριστερά από όλες τις νάρκες και ο θησαυρός πιο δεξιά.

Ως είσοδος, δίνεται ένα αρχείο στην παρακάτω μορφή (χωρίς τα σχόλια):

Συντεταγμένες σε ακεραίους (εξαίρεση η 3^η γραμμή που είναι το πλήθος διαμαντιών)

```
8 23      //συντεταγμένες εκκίνησης
130 28     //συντεταγμένες θησαυρού
156        //πλήθος διαμαντιών
23 108     //1η νάρκη
50 99      //2η νάρκη..
108 107    //3η νάρκη...
52 54
115 107...
```

Θέλουμε να υλοποιήσουμε ένα πρόγραμμα που να επιτελεί τις ακόλουθες λειτουργίες:

ΛΕΙΤΟΥΡΓΙΑ Α

Να βρίσκεται το συντομότερο μονοπάτι από το σημείο εκκίνησης και να εκτυπώνεται στην οθόνη αυστηρά στην εξής μορφή:

```
The shortest distance is 122.16872656315464
The shortest path is: (8.0,23.0)-->(56.0,23.0)-->(130.0,28.0)
```

ΛΕΙΤΟΥΡΓΙΑ Β

Ανάμεσα στα διαμάντια υπάρχει και ένα μη πολύτιμο, που ζυγίζει πιο λίγο από τα υπόλοιπα (όλα τα υπόλοιπα διαμάντια έχουν ακριβώς το ίδιο βάρος). Ευτυχώς έχουμε στη διάθεσή μας ένα ζυγό, όπως αυτός που φαίνεται στο σχήμα. Να υλοποιηθεί κώδικας που να βρίσκει το ελάχιστο πλήθος ζυγισμάτων, ώστε να εντοπίζεται το μη πολύτιμο διαμάντι. Στην οθόνη, να εκτυπώνεται το εξής μήνυμα:

```
Number of weightings: 5
```



Τρόπος υλοποίησης:

- 1) Να χρησιμοποιήσετε **Java 1.7 ή Java 1.8**. Το **μοναδικό όρισμα εισόδου** να είναι το αρχείου εισόδου.
- 2) Μπορείτε να χρησιμοποιήσετε όποιον αλγόριθμο έχετε διδαχθεί ως τμήμα του κώδικα. Θα πρέπει εσείς να εντοπίσετε ποιοι αλγόριθμοι θα πρέπει να χρησιμοποιηθούν από αυτούς που περιέχονται στην ύλη του μαθήματος. Επιτρέπεται η χρήση έτοιμου κώδικα ελεύθερα διαθέσιμου (έτοιμες υλοποιήσεις των αλγορίθμων με κατάλληλη προσαρμογή από εσάς) με την προϋπόθεση ότι θα συμπεριλάβετε σε σχόλια την πηγή του
- 3) Η **πολυπλοκότητα της Λειτουργίας Α** πρέπει να είναι υποχρεωτικά **$O(n \log n)$** στη μέση περίπτωση και της **Λειτουργίας Β** **$O(\log n)$** .
- 4) Ο πηγαίος κώδικας πρέπει να έχει **συνοπτικά σχόλια** μέσα στον κώδικα (**inline**) και **εκτενή σχόλια Javadoc** επάνω από κάθε συνάρτηση, τα οποία να εξηγούν το σκεπτικό της υλοποίησής σας και ανάλυση της πολυπλοκότητας. Τα σχόλια πρέπει να είναι **ΜΟΝΟ** στα αγγλικά ή στα ελληνικά (προσοχή στο σωστό encoding), όχι greeklish.

Παραδοτέο:

- Ένας συμπίεσμένος φάκελος που θα περιέχει αποκλειστικά α) τα αρχεία του πηγαίου κώδικα, και β) το εκτελέσιμο αρχείο jar. Ο φάκελλος θα έχει το όνομα AEM.zip και θα αποσταλεί ηλεκτρονικά από το ακαδημαϊκό σας email στο βοηθό διδασκαλίας Αθανάσιο Νάσκο (anaskos@delab.csd.auth.gr). Απορίες μπορείτε να στέλνετε και στον κ. Γούναρη.
- **Προθεσμία υποβολής:** Κυριακή 8/5/2016 11:59μμ. Θα γίνονται δεκτές ασκήσεις μέχρι και τις 10/5/2016 11:59μμ, αλλά αυτές θα έχουν μείωση βαθμού κατά 25%.

Διευκρινήσεις:

- Εκτός από τα αρχεία που αναφέρονται ως παραδοτέα, ο φάκελλος δεν πρέπει να περιέχει κάποιο άλλο αρχείο. Φοιτητές που στέλνουν netbeans, eclipse κλπ. projects ή δεν χρησιμοποιούν τον ακαδημαϊκό τους λογαριασμό **θα μηδενίζονται απευθείας**.
- Όπως αναφέρθηκε παραπάνω, επιτρέπεται η χρησιμοποίηση έτοιμου κώδικα από τρίτους με την προϋπόθεση ότι θα αναφέρεται σαφώς στα σχόλια. Όμως, όπως αναγράφεται στην ιστοσελίδα του μαθήματος, τα προγράμματα θα ελέγχονται από πρόγραμμα εντοπισμού αντιγραφών. Αν εντοπιστούν αντιγραφές μεταξύ φοιτητών, τότε οι φοιτητές θα μηδενίζονται συνολικά στο μάθημα.
- Αν δεν χρησιμοποιηθεί έτοιμος κώδικας στην υλοποίηση των αλγορίθμων, καταβάλλοντας προσωπικό κόπο για την υλοποίησή τους, τότε θα υπάρξει κάποια επιβράβευση στην βαθμολογία της εργασίας σε μορφή bonus.
- **Απαραίτητη προϋπόθεση να βαθμολογηθεί η εργασία με βαθμό >0 είναι να παράγει σωστά αποτελέσματα.**
- Στο επάνω μέρος του αρχείου με τον πηγαίο κώδικα να αναφέρονται το AEM, το όνομα και το email σας.
- Οι φοιτητές θα πρέπει να είναι έτοιμοι να δώσουν προφορικές εξηγήσεις για την υλοποίησή τους, όποτε τους ζητηθεί.
- Οι βαθμοί των ασκήσεων θα ισχύουν και για τις εξετάσεις Σεπτεμβρίου ή επί πτυχίω μέχρι να ξαναδιδαχθεί το μάθημα.

Προσομοίωση ζυγού:

Η λειτουργία του ζυγού μπορεί να προσομοιωθεί ως εξής:

```
int zygos() {
    Random randomGenerator = new Random();
    int x = randomGenerator.nextInt(100);
    if (x < 34)
        return 1; //κλίνει αριστερά
    else if (x < 67)
        return 0; //ίσο βάρος
    else
        return -1; //κλίνει δεξιά
}
```