

Robotics

Team 2



Βασική ιδέα

Ένα αυτοκινούμενο robot το οποίο θα μπορεί να βρίσκει την έξοδο άγνωστων λαβύρινθων ή να φτάνει σε κάποιο συγκεκριμένο σημείο καθορισμένο από πριν.

Στόχος

Στόχος μας είναι να προγραμματίσουμε το αυτοκίνητο να διασχίσει έναν λαβύρινθο σύμφωνα με τον αλγόριθμο Dijkstra.

Σύμφωνα με αυτόν τον αλγόριθμο το αυτοκίνητο βρίσκει τη συντομότερη διαδρομή για να κατευθυνθεί την έξοδο του λαβυρίνθου.

Υλοποίηση

Το πρόγραμμα λαμβάνει ως inputs:

- τον λαβύρινθο σε *txt* μορφή
- την αρχική θέση του του robot
- την έξοδο του λαβυρίνθου

Από το *txt* αρχείο, ο λαβύρινθος που εμπεριέχεται, μετατρέπεται σε *array* και από εκεί εντοπίζεται η αρχική του θέση.

Αξιωματικά συμβολίζεται με "@"

Υλοποίηση

Το robot ακολουθεί μια αυστηρή διαδρομή, ελέγχοντας κάθε φορά τις δυνατές του επιλογές.

1. Ελέγχει κελιά επάνω
2. Ελέγχει κελιά δεξιά
3. Ελέγχει κελιά κάτω
4. Ελέγχει κελιά αριστερά

Κάθε φορά που βρίσκει προσπελάσιμη διαδρομή, ακολουθώντας την παραπάνω ιεραρχία (1ο -> 4ο), κινεί το robot απευθείας εκεί.

Σε περίπτωση που δεν υπάρχει προσπελάσιμη διαδρομή (είτε τοίχος είτε προήλθε από προηγούμενο κελί), επιστρέφει στον αμέσως προηγούμενο κόμβο.

Υλοποίηση

Στο κελί που βρίσκομαι τώρα ελέγχω πόσα προσπελάσιμα γειτονικά κελιά έχω. Εάν ο αριθμός τους είναι πάνω από 1 τότε δημιουργώ κόμβο στο κελί αυτό. Κάθε φορά που βρίσκεται σε κελί με παραπάνω από μία δυνατές διαδρομές, δημιουργεί έναν επιπλέον κόμβο και κινούμαστε κατά μια θέση. Στην καινούργια θέση ελέγχουμε πόσες διαθέσιμες διαδρομές έχουμε, και αν είναι 2 ή περισσότερες τότε σε εκείνη την θέση δημιουργώ κόμβο και συνεχίζω.

Υλοποίηση

Αφού ανακαλύψει κάθε πιθανή διαδρομή, το πρόγραμμα τερματίζει και εκτυπώνει:

- Τον αρχικό άγνωστο λαβύρινθο
- Το λαβύρινθο με όσα κελιά είναι δυνατό να ανακαλυφθούν και τους κόμβους που δημιουργήθηκαν
- Τον αριθμό κόμβων
- Το μητρώο γειτνίασης
- Την καλύτερη διαδρομή (αριθμητικά)
- Τον λαβύρινθο με την καλύτερη διαδρομή highlighted

Επειδή στα λόγια ίσως είναι λίγο ακαταλαβίστικο, το επόμενο gif συνοψίζει πρακτικά την όλη διαδικασία...

Υλοποίηση

```
## ## ## ## ##
## O ## ## ##
## @ - - - ##
## # - - - ##
## - - - ##
## - ## ## ##
## - ## ## ##
## - - - - ##
## ## ## ## ##
```

We have 1 points

Distance is 1