

Πανεπιστήμιο Ιωαννίνων Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Αλγόριθμοι και Πολυπλοκότητα

2η Εργασία

Δημιουργία λαβυρίνθου, σχεδίαση διαδρομής από την είσοδο στην έξοδο

Όνομα: Δημήτριος

Επώνυμο: Κάτσανος

ΑΜ:2281

Email:int02281@uoi.gr

Πίνακας περιεχομένων

1. Υλικό που χρησιμοποιήθηκε
2. Λογισμικό που χρησιμοποιήθηκε
3. Περίληψη στη δομή ξένων συνόλων
4. Εισαγωγή στο πρόβλημα του λαβυρίνθου
5. Αποτελέσματα
6. Συμπεράσματα

1. Υλικό που χρησιμοποιήθηκε

Το υλικό που χρησιμοποιήθηκε για τα πειράματα είναι:

- Τύπος συστήματος: Λειτουργικό σύστημα 64 bit, επεξεργαστής τεχνολογίας x64
- Επεξεργαστής: Intel Core i5 2.70GHz
- Εγκατεστημένη RAM:8,00GB DDR3 (798MHz)

2. Λογισμικό που χρησιμοποιήθηκε

Το λογισμικό που χρησιμοποιήθηκε είναι:

- Python 3.8.10
- Visual Studio Code

3. Περίληψη

Μια δομή ξένων συνόλων είναι μια δομή δεδομένων η οποία χειρίζεται ένα σύνολο στοιχείων τα οποία διαχωρίζονται σε ένα αριθμό μη επικαλυπτόμενων υποσυνόλων. Σε αυτή την τεχνική αναφορά εξετάζουμε το πρόβλημα της δημιουργία κα επίλυσης λαβυρίνθων. Η δημιουργία ενός λαβυρίνθου υλοποιείται με την δομή των ξένων συνόλων, ενώ η επίλυση τους επιτυγχάνεται με πολλούς διαφορετικούς αλγορίθμους, όπως αναζήτησης πρώτα κατά πλάτος(Breadth First Search, BFS)

4. Εισαγωγή

Το πρόβλημα του λαβύρινθου (maze) μπορεί να θεωρηθεί ως ένα πλέγμα $M \times N$ το οποίο αποτελείται από κελιά με κάθε κελί να έχει 4 πλευρές κάποιες από τις οποίες ενδεχομένως να είναι ανοιχτές. Τα κελιά που είναι στην εξωτερικά είναι όλα κλειστά εκτός από την είσοδο και την έξοδο του λαβυρίνθου. Ο λαβύρινθος είναι έγκυρος μόνο όταν υπάρχει μια διαδρομή η οποία να ξεκινάει από μία είσοδο να δημιουργείται από τις ανοιχτές πλευρές μεταξύ των κελιών και να καταλήγει σε μία έξοδο.

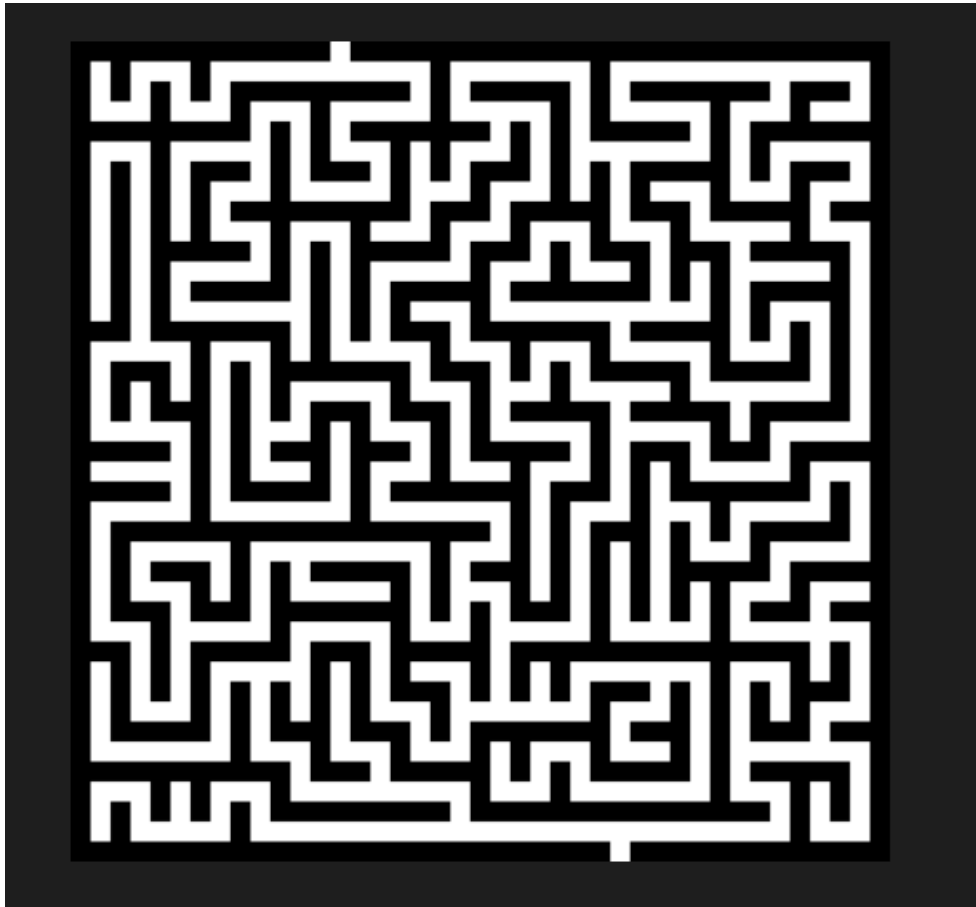
5. Αποτελέσματα

Ο λαβύρινθος δημιουργείται με την δομή ξένων συνόλων (disjoint-set) λαμβάνοντας ως είσοδο τις διαστάσεις του πλέγματος (πλάτος και ύψος), και (αν ο χρήστης θέλει) τη θέση εισόδου και τη θέση εξόδου, ενώ αν δεν θέλει ορίζεται αυτόματα. Οι θέσεις αυτές προσδιορίζονται από την εξωτερική πλευρά του λαβυρίνθου (UP = πάνω, DOWN = κάτω, LEFT = αριστερά, RIGHT = δεξιά) και με ένα αριθμό που προσδιορίζει το κελί.

Τα δεδομένα που εισάγει ο χρήστης είναι τα εξής:

```
PS C:\Users\user\Documents\KATΣΑΝΟΣ_2281_2ΕΡΓΑΣΙΑ> python -u maze.py
Enter a number height for the Width: 20
Enter a number height for the maze: 20
Give the start num position for the maze: 5
Give the start position for the maze: up
Give the end num position for the maze: 70
Give the end position for the maze: RIGHT
iterations: 2747
been in deadend: 405
```

Έπειτα, ο λαβύρινθος εμφανίζεται όπως φαίνεται παρακάτω:



Τέλος, η λύση του λαβυρίνθου, ο εντοπισμός δηλαδή της διαδρομής από την είσοδο στην έξοδο επιτυγχάνεται με δύο τρόπους, με τον αλγόριθμο αναζήτησης πρώτα κατά πλάτος (Breadth First Search, BFS) Υπάρχει και άλλος ένας αλγόριθμος που επιλύει έναν λαβύρινθο, ο A* ο οποίος είναι κατά βάση BestFS αλλά και με τον αλγόριθμο αναζήτησης πρώτα κατά βάθος (Depth First Search, DFS) αλλά και με ευριστική συνάρτηση. Στον κώδικα ο λαβύρινθος εμφανίζεται με τον αλγόριθμο του bfs

6. Συμπεράσματα

Μέσα από αυτή την εργασία κατανοούμε καλύτερα το πώς να χρησιμοποιούμε την δομή των ξένων συνόλων για να δημιουργήσουμε τον λαβύρινθο. Επίσης κατανοούμε τον αλγόριθμο αναζήτησης