

---

# Δημιουργία λαβυρίνθου, σχεδίαση διαδρομής από την είσοδο στην έξοδο

---

Σταυροπούλου Μαρία

ΑΜ: 2058

23 Δεκεμβρίου 2021

## Περίληψη

Μια δομή ξένων συνόλων (disjoint-set data structure) 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 δημιουργίας και 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), αναζήτηση πρώτα κατά βάθος (Depth First Search, DFS)

## 1 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.

## 2 Αποτελέσματα

Για την συγγραφή του κώδικα χρησιμοποιήθηκε η Python 3.9.6 και το Visual Studio Code. Τα χαρακτηριστικά του υπολογιστή είναι i3-4000M(2.40GHz), 4 GB DDR3. Για την οπτικοποίηση των αποτελεσμάτων χρησιμοποιήθηκε η βιβλιοθήκη matplotlib.

Ο λαβύρινθος δημιουργείτε 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.

Παρακάτω φαίνονται δύο παραδείγματα με τα δεδομένα που δίνει ο χρήστης.

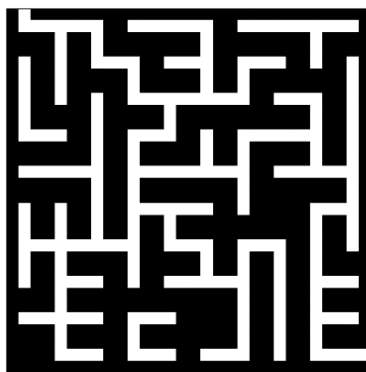
```
Give length for the maze: 10
Give height for the maze: 10
Do you want to choose the entrance and exit for the maze?(Please answer with Yes or No) yes
Give the start num position for the maze: 3
Give the start position for the maze: UP
Give the end num position for the maze: 90
Give the end position for the maze: LEFT
```

Εικόνα 1: Παράδειγμα δεδομένων που δίνει ο χρήστης είσοδο και έξοδο

```
Give length for the maze: 10
Give height for the maze: 10
Do you want to choose the entrance and exit for the maze?(Please answer with Yes or No) no
```

Εικόνα 2: Παράδειγμα δεδομένων που δεν δίνει ο χρήστης είσοδο

Έπειτα, ο λαβύρινθος σχεδιάζεται με την βοήθεια της βιβλιοθήκης matplotlib και εμφανίζεται όπως φαίνεται παρακάτω.



Εικόνα 3: Παράδειγμα εμφάνισης λαβυρίνθου 10 x 10

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) και με τον αλγόριθμο αναζήτησης πρώτα κατά βάθος (Depth First Search, DFS). Υπάρχει και άλλος ένας αλγόριθμος που επιλύει έναν λαβύρινθο, ο A\* ο οποίος είναι κατά βάση BestFS αλλά με ευριστική συνάρτηση. Στον κώδικα ρωτάμε το χρήστη με ποιον από τους δύο αλγορίθμους (bfs,dfs) θέλει να λυθεί ο λαβύρινθος και αν λύνεται σωστά εμφανίζεται στην οθόνη True.

Παρακάτω δίνεται ένα παράδειγμα τα αποτελέσματα που εμφανίζονται στον χρήστη.

```
How do you want to solve the maze? (with bfs or dfs algorithm) bfs
True
```

Εικόνα 4: Παράδειγμα αποτελεσμάτων που εμφανίζονται στο χρήστη.

### 3 Συμπεράσματα

Μέσα από αυτή την εργασία κατανοούμε καλύτερα το πώς να χρησιμοποιούμε την δομή των ξένων συνόλων για να δημιουργήσουμε τον λαβύρινθο. Επίσης, κατανοούμε τους αλγόριθμους αναζήτησης και τις κύριες διαφορές αυτών των δύο αλγορίθμων.