

Εργασία 3 - Αρχές Γλωσσών Προγραμματισμού 2016-2017

Μέλη Ομάδας

Ελευθέριος Καραμπάς - AM:1115201400064

Θανάσης Φιλλιπίδης - AM:1115201400215

Για την υλοποίηση της εργασίας εργαστήκαμε ως εξής: Αναπαραστήσαμε τον λαβύρινθο σαν ένα γράφο ο οποίος αποτελείται από κόμβους που είναι τα κελιά(cells) και για ακμές των κελιών τους τοίχους που έχουμε αφαιρέσει.

Κάθε κελί με συντεταγμένες (x,y) έχει ένα μοναδικό id το οποίο υπολογίζεται απο την πράξη $id = y * width + x$ όπου width είναι το πλάτος του λαβυρίνθου.

Για το **Ερώτημα 3** χρησιμοποιήσαμε την Αναζήτηση Κατα Βάθος (DFS) χωρίς να κρατάμε τα κελιά που έχουμε επισκεφτεί. Καθώς ακολουθούμε ένα μονοπάτι αν φτάσουμε στη λύση επιστρέφεται προς τα πίσω μία λίστα, με πρώτο στοιχείο ένα tuple $(-1,-1)$ το οποίο χρησιμοποιούμε ως flag για να ενημερώσουμε την αναζήτηση μην προχωρήσει άλλο αφού έχουμε βρεί λύση ώστε να επιστρέψει το μονοπάτι αυτό. Στην περίπτωση που ένα μονοπάτι φτάσει σε αδιέξοδο επιστρέφεται η λίστα με πρώτο στοιχείο το tuple $(-2,-2)$ το οποίο χρησιμοποιείται ως flag για να ενημερώσουμε την αναζήτηση να διαλέξει κάποιο άλλο μονοπάτι προς εξερεύνηση.

Για το **Ερώτημα 4** το μονοπάτι που δέχεται η showMaze, μία λίστα απο τις συντεταγμένες των κελίων, το μετατρέπουμε σε id του κάθε κελιού με τον τρόπο που είπαμε παραπάνω και αυτη τη λίστα των κελίων την ταξινομούμε κατα αύξουσα id. Με τη ταξινομημένη αυτή λίστα και το Maze που έχουμε δημιουργούμε την αναπαράσταση του λαβυρίνθου μας.

Για το **Ερώτημα Bonus** χρησιμοποιήσαμε τον αλγόριθμο της Αναζήτησης Κατά Πλάτος ενός γράφου (BFS) με βοηθητικές δομές το explored set το οποίο κρατάει ποιά κελιά έχουμε ήδη εξερευνήσει και τη δομή frontier (Queue) που κράταει ποιοί είναι οι επόμενοι κόμβοι προς εξερεύνηση αλλά και το μονοπάτι που έχουμε ακολουθήσει μέχρι εκεί για κάθε κόμβο.