3η Εργασία - ΘΠ01: Αρχές Γλωσσών Προγραμματισμού
ΟΜΑΔΙΚΗ ΕΡΓΑΣΙΑ

ΣΥΜΜΕΤΕΧΟΝΤΕΣ:

ΡΙΤΣΟΓΙΑΝΝΗ ΑΡΓΥΡΩ Α.Μ. 1115201400171
ΤΡΙΑΝΤΑΦΥΛΛΟΥ ΛΕΩΝΙΔΑΣ Α.Μ. 1115201400202

Ο φάκελος αυτός περιέχει 1 αρχείο το maze.hs.
Η εργασία αυτή έχει υλοποιηθεί σύμφωνα με τις απαιτήσεις της εκφώνησης.

Perfect Maze

Αλγόριθμος Κατασκευής
Για αυτόν τον λαβύρινθο έχει υλοποιηθεί ο αλγόριθμος κατασκευής (σύμφωνα με τον αλγόριθμο kruskal που έχει δοθεί και η αναπαράσταση των δομών, set και join_set, με λίστα ακεραίων που αναπαρηστούν τα κελία, σε αύξοντα αριθμό, του λαβυρίνθου) και ο αλγόριμθος επίλυσης (σύμφωνα με τον αλγόριθμο της κατά βάθους αναζήτησης , DFS).

Αλγόριθμος Επίλυσης:
Σε κάθε κελί που είμαστε, βρίσκουμε τους γείτονες με μια σειρά (π.χ. αν υπάρχουν όλοι οι γείτονες: πάνω,αριστερά,δεξιά ,κάτω) και ξεκινάμε για κάθε γείτονα με τη σειρά να κάνουμε την ίδια διαδικασία. Παράλληλα προσθέτουμε στο μονοπάτι επίλυσης κάθε κελί που επισκεπτόμαστε. Αν φτάσουμε σε αδιέξοδο, τότε γυρνάμε ένα βήμα πίσω και πάιρνουμε τον επόμενο γείτονα αν υπάρχει. Αν δεν υπάρχει γυρνάμε άλλο ένα βήμα πίσω κ.λπ. . Όταν ο αλγόριθμος γυρνάει πίσω το μονοπάτι επίλυσης έχει μόνο τα κελιά μέχρι το σημείο που βρισκόμαστε χωρίς δηλαδή αυτά του αδιεξόδου.

Για να τρέξει το πρόγραμμα/λαβύρινθο αυτό χρησημοποιήστε τις συναρτήσεις:

- runmaze w h , η οποία εκτυπώνει μόνο έναν τέλειο λαβύρινθο (Προσοχή: όπου w:πλάτος και h:ύψος).
- solvemaze w h (sx,sy) (gx,gy), η οποία εκτυπώνει έναν τέλειο λαβύρινθο μαζί με την λύση του (όπου w:πλάτος,h:ύψος,(sx,sy):συντεταγμένες αρχικού κελιού και όπου sx:ύψος,sy:πλάτος και (gx,gy): συντεταγμένες τελικού κελίου και όπου gx:ύψος,gy:πλάτος)

Braid Maze - Bonus

Για τον λαβύρινθο αυτό έχει υλοποιηθεί ο αντίστοιχος αλγόριθμος κατασκευής μόνο που στην περίπτωση αυτή τα κελία που έχουν αδιέξοδο ενώνονται με τα γειτονικά κελιά τους έτσι ώστε να δημιουργηθεί λαβύρινθος που οδηγεί σε ένα κελί από πολλαπλές διαδρομές(δημιουργόντας πολλές φορές και κύκλους). Για τα κελιά που δεν βρίσκονται στην τελευταία στήλη ή τελευταία γραμμή βρίσκουμε τους γείτονες τους και αν έχουν μόνο ένα γείτονα τότε βγάζουμε τον δεξιά ή τον κάτω τοίχο. Για την τελευταία στήλη αν υπάρχει ένας γείτονας τότε βγάζουμε τον κάτω τοίχο ή πάμε στο αριστερά κελί και ρίχνουμε τον δεξιά τόιχο. Για την τελευταία γραμμή βγάζουμε τον δεξιά τοίχο ή πάμε στο από πάνω κελί και βγάζουμε τον κάτω τοίχο του. Για το (0,0) και το (h-1,w-1) κελί (δηλαδή τα δύο γειτονικά) αφαιρούμε αντίστοιχα όλους τους τοίχους

Για την επίλυση του braid έχει χρησιμοποιηθεί πάλι η DFS απλά στην περίπτωση αυτή κρατάμε τα κελιά που έχουμε επισκεφθεί ώστε να μην τα ξαναεπισκεφθούμε. Όπως έχει υλοποιηθεί ο

αλγόριθμος βρίσκει πολλές λύσεις, ωστόσο κρατείται και παρουσιάζεται η πρώτη λύση που βρίσκεται.

Για να εκτυπώσετε τον braid λαβύρινθο χρησημοποιήστε τη συνάρτηση:
- printbraid w h
(όπου w:πλάτος,h:ύψος)

Για να εκτυπώσετε τον braid λαβύρινθο μαζί με τη λύση του χρησημοποιήστε τη συνάρτηση:
- solvebraid w h (sx,sy) (gx,gy)
(όπου w:πλάτος,h:ύψος,(sx,sy):συντεταγμένες αρχικού κελιού και όπου sx:ύψος,sy:πλάτος και (gx,gy): συντεταγμένες τελικού κελίου και όπου gx:ύψος,gy:πλάτος)

Εκτύπωση και λύση και των δύο λαβυρίνθων:

Για να εκτυπώσετε και τους δύο λαβυρίνθους μαζί χρησημοποιήστε τη συνάρτηση:
- printboth w h
(όπου w:πλάτος,h:ύψος)

Για να εκτυπώσετε και τους δύο λαβυρίνθους μαζί με τη λύση του καθενός χρησημοποιήστε τη συνάρτηση:
- solveboth w h (sx,sy) (gx,gy)
(όπου w:πλάτος,h:ύψος,(sx,sy):συντεταγμένες αρχικού κελιού και όπου sx:ύψος,sy:πλάτος και (gx,gy): συντεταγμένες τελικού κελίου και όπου gx:ύψος,gy:πλάτος)

Ιστοσελίδες που βοήθησαν στην υλοποίηση της εργασίας (www.google.com, www.stackoverflow.com)