

# Δημιουργία λαβυρίνθου

Κώστας Άγγελος

18 Δεκεμβρίου 2022

## Περίληψη

Η συγκεκριμένη εργασία λύνει το πρόβλημα σχεδίασης ενός λαβυρίνθου σύμφωνα με τον αλγόριθμο δημιουργίας ο οποίος κάνει χρήση της δομής δεδομένων disjoint sets. Εξηγείται ο τρόπος εκτέλεσης του προγράμματος και παρουσιάζονται ενδεικτικά samples από την έξοδο του.

## 1. Εισαγωγή

Το πρόβλημα το οποίο καλούμαστε να λύσουμε στην συγκεκριμένη εργασία είναι η δημιουργία ενός τυχαίου λαβυρίνθου ο οποίος έχει πάντα μια λύση και δεν σχηματίζει κάποιο κύκλο. Υπάρχουν αρκετοί τρόποι οπτικοποίησης του λαβυρίνθου, στην περίπτωσή μας εμφανίζεται με ASCII χαρακτήρες. Για την εύρεση της λύσης έγινε χρήση του αλγορίθμου BFS ο οποίος λειτουργεί με την λογική FIFO. Η έξοδος του προγράμματος μας δίνει την αρχική σχεδίαση του λαβυρίνθου ( μόνο τοίχους στις διαστάσεις που πληκτρολογήσαμε ), την τελική με τοίχους και τα κενά προς την λύση και τέλος την διαδρομή που βρίσκει ο αλγόριθμος BFS.

## 2. Πειραματικά αποτελέσματα

Για την καταγραφή των πειραματικών αποτελεσμάτων χρησιμοποιήθηκε το εξής setup :

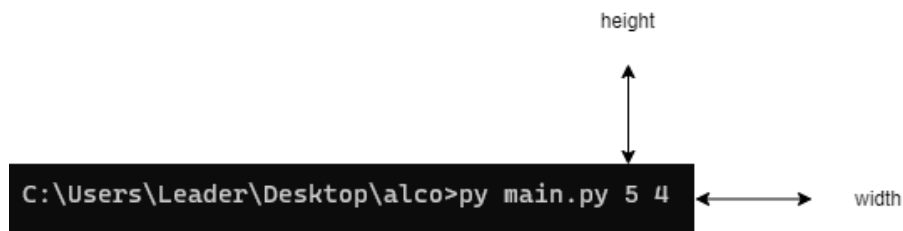
- IDE : PyCharm 2022.2.3
- Γλώσσα προγραμματισμού : Python 3.10.8

Χαρακτηριστικά υπολογιστή :

- CPU : AMD Ryzen 7 5800x
- RAM : 16 GB DDR4 (3200MHz)

## 3. Οδηγίες εκτέλεσης και αποτελέσματα

Για να τρέξουμε τον κώδικα αφότου μεταφερθούμε στον σωστό κατάλογο πρέπει να εκτελέσουμε την εξής εντολή. Ο αλγόριθμος από default θα βάλει την αρχική θέση του αλγορίθμου στο κελί 1 και την έξοδο στο κελί  $height * width$ , δηλαδή για  $height=5$  και  $width=4$  η έξοδος θα είναι στο κελί 20.



Αρχικός λαβύρινθος για height=5 και width=4 :

```
*** Initial Labyrinth ***  
  
+-----+  
|       |  
+       +  
|       |  
+       +  
|       |  
+       +  
|       |  
+-----+
```

Τελικός λαβύρινθος για τις ίδιες διαστάσεις :

```
*** Final Labyrinth ***  
  
+-----+--+  
|       |  |  
+  --+--+  +  
|       |  |  
+---  +  +  +  
|       |  |  
+--+  +  --+  
|  |  |  |  
+--+  +--+  +  
|  |  |  |  
+--+--+--++
```

Το τελικό path που βρίσκει ο BFS :

```
*** Path to solution using bfs ***  
  
1 --> 5 --> 6 --> 10 --> 11 --> 15 --> 16 --> 20
```

## 4. Συμπεράσματα

Από τις παραπάνω εξόδους μπορούμε να δούμε ότι ο αλγόριθμος BFS μπορεί να βρει λύση και μας λύνει το πρόβλημα. Για περαιτέρω έρευνα θα μπορούσαμε να υλοποιήσουμε περισσότερους αλγορίθμους αναζήτησης όπως ο DFS, Dijkstra και να δούμε πως αυτοί αποδίδουν σε θέμα χώρου και χρόνου στο πρόβλημα σχεδίασης και λύσης του λαβυρίνθου.