

## Εργασία 1

Δημοσθένης Θεοδοσίου (1115202300051)

Οκτώβριος 2024

### Άσκηση 1:

Ουσιαστικά στην άσκηση 1 ακολουθούμε τον ψευδοκώδικα που μας δόθηκε από το φροντιστήριο. Το μόνο ιδιαίτερο που κάνουμε είναι πως το fringe είναι stack, για να μπαίνουν τα στοιχεία σωστά στο fringe.

### Άσκηση 2:

Ακριβώς το ίδιο με παραπάνω με μόνη διαφορά ότι το fringe είναι Queue.

### Άσκηση 3:

Πάλι ακριβώς το ίδιο με παραπάνω με την διαφορά ότι το fringe είναι PriorityQueue και κάθε στοιχείο ταξινομείται στην λίστα με βάση το ελάχιστο κόστος που χρειάστηκε για να επισκεπτούμε κάθε κόμβο.

### Άσκηση 4:

Περίπου η ίδια υλοποίηση με το UCS, απλά το κόστος πλέον είναι το πραγματικό κόστος συν το ευρετικό. Επίσης αν επισκεπτούμε έναν κόμβο που έχουμε ξαναδεί δεν τον αγνοούμε όπως στις άλλες περιπτώσεις, αλλά εξετάζουμε αν φτάσαμε εκεί με χαμηλότερο κόστος από ότι πριν και αν ναι το ανανεώνουμε στο visited.

### Άσκηση 5:

- `getStartState`: Σε κάθε καινούριο state επιστρέφω tuple που περιέχει ένα tuple με το ποιές κορυφές έχουμε επισκεπτεί και τις συντεταγμένες του παχμαν.
- `isGoalState`: Σε αυτή την συνάρτηση ελέγχουμε αν ο παχμαν έχει επισκεπτεί όλες τις κορυφές ή όχι.
- `getSuccessors`: Εδώ κάνουμε σχεδόν το ίδιο με την υλοποίηση της συνάρτησης στο `SearchPositionProblem` με την αλλαγή, ότι για κάθε καινούριες συντεταγμένες που παράγουμε, ελέγχουμε εάν είναι σε μία από τις κορυφές του grid. Εάν είναι τότε τροποποιούμε το tuple με το ποιες κορυφές έχουμε επισκεπτεί. Πρέπει να σημειωθεί πως στο tuple δεν μπορούμε να κάνουμε απλά edit τα δεδομένα έτσι απλά, αφού δεν το επιτρέπει η python. Για αυτό τον λόγο μετατρέπουμε το tuple σε λίστα, κάνουμε edit την λίστα και την ξαναμετατρέπουμε σε tuple. Αφού τα κάνουμε όλα αυτά απλά περνάμε το tuple με τις νέες συντεταγμένες και το νέο tuple των visited corners ως καινούριο state.

### Άσκηση 6:

Για την ευρετική του `CornersProblem` υπολογίζουμε το άθροισμα των ευκλείδειων αποστάσεων για όλες τις κορυφές που ακόμη δεν έχουμε επισκεπτεί και το διαιρούμε με 1.5 για να παραμένει admissible.

### Άσκηση 7:

Στην συγκεκριμένη ευρετική για όλα φαγητά τρέχουμε `manhattanDistance` και βρίσκουμε το μέγιστο `manhattanDistance` από τον πάχμαν προς κάποιο φαγητό. Έπειτα για όλα τα φαγητά εκείνα που κατά `manhattan` απέχουν το μέγιστο `manhattan distance` που βρήκαμε πριν, τρέχουμε `maze distance` και επιστρέφουμε το μεγαλύτερο. Εδώ η ευρετική δεν είναι trivial, διότι τρέχουμε ελάχιστες φορές το `mazedistance` κάθε φορά.

## Άσκηση 8:

Για να βρούμε το πιο κοντινό φαγητό, ουσιαστικά απλά τρέχουμε bfs μέχρις ότου να συναντήσουμε το πρώτο φαγητό.