

Project #1

Student name: *Stylianos Bitzas (sma1100202)*

Course: *Artificial Intelligence (ΥΣ02)* – Professor: *Manolis Koubarakis*
Due date: *November 5th, 2019*

Question 1

Να κάνετε το Project 1 από τα Pacman projects (<http://ai.berkeley.edu/tutorial.html>).

Answer. Γενικά έχω συμπεριλάβει αναλυτικά σχόλια σε όλο τον κώδικα που έχω γράψει θα σχολιάσω μόνο τη δόμη της λύσης και κάποιες λεπτομέρειες για τις αποφάσεις που πήρα.

- **Για τα ερωτήματα Q1-Q4** χρησιμοποίησα ένα ενιαίο αλγόριθμο `graphSearch` που χρησιμοποιεί μία απλή δομή `tree`, που είναι ένα δέντρο όπου αποθηκεύεται σε ένα `dictionary`, και μία κλάση `node` που έχει τις απαραίτητες μεθόδους για την υλοποίηση των αλγορίθμων. Για το σύνορο έχω χρησιμοποιήσει μία κλάση που την έχω ονομάσει `fringe` για τους `dfs`, `bfs`, `ucs` και την επέκταση της `fringeHeur` για τον `astar`. Αυτή η κλάση χρησιμοποιεί τις έτοιμες δομές στο αρχείο `util.py` και στην ούσια απλά επιλέγει την κατάλληλη δομή κάθε φορά και/ή επεκτείνει κάποιες μεθόδους.
- **Για το ερώτημα Q5** κρατάω μία λίστα/πλειάδα που περιέχει τις τιμές 0 για τις 4 γωνίες και όταν βρεθεί ο `pacman` σε μία γωνία παίρνει την τιμή 1. Ο στόχος επιτυγχάνεται όταν ο `pacman` επισκεφθεί για πρώτη φορά και τις 4 γωνίες.
- **Για το ερώτημα Q6** χρησιμοποιώ την έτοιμη συνάρτηση `manhattanDistance` του `util.py` για να φτιάξω την `heuristic` που δίνει την μεγίστη απόσταση μεταξύ του `pacman` και τις ενεργές γωνίες καθώς θα πρέπει τουλάχιστον να πάει μέχρι εκείνη την ενεργή γωνία πριν φτάσουμε σε κατάσταση στόχου.
- **Για το ερώτημα Q7** χρησιμοποιώ την έτοιμη συνάρτηση `manhattanDistance` του `util.py` για να φτιάξω την `heuristic` που δίνει την μεγίστη απόσταση μεταξύ του `pacman` και των κουκίδων που απομένουν να φάει ο `pacman` καθώς και την μεγίστη απόσταση μεταξύ των κουκίδων αυτών και παίρνω το `maximum` αυτών των δύο κάθε φορά καθώς εάν έχω δύο ευρετικές συναρτήσεις (παραδεκτές και συνεπείς) h_1, h_2 τότε το $\max(h_1, h_2)$ είναι ευρέτικη και μάλιστα καλύτερη από τις h_1 και h_2 . Για να πάρω την έξτρα μονάδα βρήκα μία λύση που χρησιμοποιεί την συνάρτηση `mazeDistance` αλλά αυτή η ευρετική μετά παρολό που κάνει `expand` λιγότερους κόμβους από την πρώτη (περίπου 10% λιγότερους) ο χρόνος εκτέλεσης είναι διπλάσιος (20 δευτερόλεπτα περίπου) οπότε έπελεξα να μην το συμπεριλάβω καθώς δεν θέωρω ότι προσφέρει κάτι ουσιαστικό αφού εξάλλου κάθε φορά η `mazeDistance` τρέχει `bfs`.
- **Για το ερώτημα Q8** δεν υπάρχει κάτι ιδιαίτερο να σχολιάσω.