# Τεχνητή Νοημοσύνη

## 1115201800006_project1_pacman.pdf

Αναγνωστόπουλος Αθανάσιος

AM: 1115201800006

### Question 1

Για την υλοποίηση του αλγορίθμου Depth First Search (DFS) χρησιμοποίησα στοίβα για την υλοποίηση του συνόρου frontier και την δομή set της python για το εξερευνημένο σύνολο (Ex).

Αλγόριθμος από σελ. 11 pdf ‘’Αναζήτηση σε Γράφους’’.

### Question 2

Για την υλοποίηση του αλγορίθμου Breadth First Search (BFS) χρησιμοποίησα ουρά για την υλοποίηση του συνόρου frontier και την δομή set της python για το εξερευνημένο σύνολο (Ex).

Αλγόριθμος από σελ. 27 pdf ‘’Αναζήτηση σε Γράφους’’.

### Question 3

Για την υλοποίηση του αλγορίθμου Uniform Cost Search (UCS) χρησιμοποίησα ουρά προτεραιότητας ταξινομημένη με βάση το κόστος για την υλοποίηση του συνόρου frontier. Έτσι κάθε φορά που γίνεται pop από την ουρά παίρνουμε τον κόμβο με το λιγότερο κόστος, με αποτέλεσμα να έχουμε την βέλτιστη λύση στο τέλος.

Για το εξερευνημένο σύνολο (Ex) χρησιμοποίησα την δομή set της python.

Αλγόριθμος από σελ. 37 pdf ‘’Αναζήτηση σε Γράφους’’.

## Question 4

Για την υλοποίηση του αλγορίθμου A* χρησιμοποίησα τον ίδιο κώδικα με τον αλγόριθμο UCS. Η μόνη διαφορά είναι πως το κόστος είναι g(n) στον UCS, δηλαδή το κόστος που χρειάζεται να φτάσουμε έως τον κόμβο n, ένω ο A* χρησιμοποιεί το κόστος g(n) + h(n), δηλαδή συνυπολογίζει και το κόστος από τον κόμβο μέχρι τον στόχο.

Ο υπολογισμός του h(n) γίνεται με μία ευρετική συνάρτηση.

## Question 5

Στο πρόβλημα Finding All The Corners πρέπει να συμπληρωθεί η κλάση CornersProblem. Αρχικά το state αποτελείται από την θέση του pacman (state[0]) αλλά και από το πόσες γωνίες έχουν μείνει για επίσκεψη (state[1]). Το Goal State θα είναι όταν έχουν επισκεφθεί όλες οι γωνίες, δηλαδή όταν το state[1] δεν περιέχει στοιχεία. Επίσης κάθε φορά στη συνάρτηση getSuccessors ελέγχεται αν επισκέφτηκε κάποια γωνία ώστε να αφαιρεθεί από την λίστα με τις γωνίες που δεν έχουν επισκεφθεί.

## Question 6

Στο πρόβλημα Corners Problem: Heuristic χρησιμοποίησα την απόσταση Manhattan για την ευρετική συνάρτηση. Για όλες τις γωνίες που δεν έχουμε επισκεφθεί η γωνία με την μεγαλύτερη

απόσταση Manhattan θα είναι και η γωνία που θα επισκεφθούμε τελευταία, δηλαδή και το Goal State.

## Question 7

Στο πρόβλημα Eating All The Dots χρησιμοποίησα την ευρετική από το προηγούμενο ερώτημα (Question 6) με την διαφορά πως πλέον έχουμε λίστα με foods και όχι λίστα με corners.

## Question 8

Στο πρόβλημα Suboptimal Search συμπλήρωσα το Goal State της κλάσης AnyFoodSearchProblem ελέγχοντας αν το pacman έφτασε σε κάποιο food. Στη συνέχεια στη συνάρτηση findPathToClosestDot χρησιμοποιώ bfs όπου επιστρέφει μια λίστα από Actions στο πιο κοντινό food.