Εργασια 1 Τεχνητης Νοημοσυνης - Τουρνής Ιωάννης ΑΜ:2000192

-- Αρχειο ReadMe.txt
-- Το παρων αρχειο περιεχει κυριως γενικα σχολια που ειναι απαραιτητα για την κατανοηση του κωδικα μου και σχεδιαστικες επιλογες.
-- Τα αρχεια που εχουν πειραχτει ειναι τα search.py , searchAgents.py

-- Question 1

Υλοποιειται η Fixed Food Dot με την χρηση Depth First Search (DFS) αναζητησης.
Αρχικα δημιουργουμαι μια stack και εναν πινακα visited που δειχνει ποιους κομβους εχουμε επισκεφτει .
Χρησιμοποιω μια while η οποια ισχυει οσο η στοιβα δεν ειναι αδεια , αφαιρει καθε φορα το πανω πανω στοιχειο και υπαρχουν 2 ενδεχομενα :
- Α) Να ειναι Goal-State οποτε φτασαμε στον ζητουμενο κομβο και επιστρεφουμε την διαδρομη που καναμε
- Β) Να μην ειναι Goal-State , και αμα δεν εχουμε επισκεφτει τον κομβο τον βαζουμε στην visited. Επειτα βρισκουμαι ολους τους γειτονες του current-node που ειμαστε τωρα και τους προσθετουμε στην στοιβα για να εξεταστουν.

Η υλοποιοηση του DFS γινεται με στοιβα που ειναι LIFO ετσι ωστε να γινεται αναζητηση σε ΒΑΘΟΣ.

-- Question 2

Υλοποιειται αναζητηση παρομοια με το ερωτημα 1 αλλα με την χρηση Breadth First Search (BFS) αναζητησης.
Και γιαυτον τον λογο χρησιμοποιω ακριβως τον ιδιο κωδικα με απο πανω με την μονη διαφορα οτι αντι για χρηση Στοιβας (Stack) η υλοποιηση γινεται με Ουρα (Queue) η οποια ειναι FIFO ετσι ωστε να γινεται αναζητηση σε ΠΛΑΤΟΣ.

-- Question 3

Ουσιαστικα ο αλγοριθμος που υλοποιησα χρησιμοποιει την BFS υλοποιηση απο το ερωτ.2 και ψαχνει να βρει τον Goal-State node μεσα απο τα διαφορα μονοπατια τα οποια μπορει να προκειψουν. Κραταω εναν πινακα με το κοστος που χρειαζομαι για να φτασω σε εναν κομβο Χ , προσθετοντας στον συγκεκριμενο κομβο το κοστος που εχω μετρησει ηδη απο τους κομβους που περασα για να φτασω στον Χ + το κοστος του node(Χ-1) -> Χ .

-- Question 4

Ιδια λογικη υλοποιησης με την μονη διαφορα οτι χρησιμοποιειται PriorityQueue και χρησιμοποιειται huristic. Εσωτερικα της FOR που βρισκοται οι Successors ενος node , υπολογιζουμαι το κοστος του μονοπατιου που δημιουργειται απο τις κινησεις του pacman μεχρι το κομβο αυτο . Η κυρια διαφορα εδω ειναι οταν γινεται PUSH ενας νεος κομβος , διοτι σαν κοστος μονοπατιου δινεται το κοστος του μονοπατιου + το κοστος που μας επιστρεφει η Huristic για τον συγκεκριμενο node.

-- Question 5

CornersProblem in searchAgents.py
Το προβλημα αυτο αφορα την επισκεψει του pacman στις 4 γωνιες της πιστας και να φαει το Food που βρισκεται εκει.

- Goal-State : Θεωρησα την περιπτωση που οι 4 γωνιες θα βρισκονται μεσα στον πινακα self.corners και προφανως οτι ο πινακας visited θα εχει ακριβως 4 στοιχεια , δηλαδη τις γωνιες που ζηταμε.

- GetSuccessors : Ως Successors απο εναν node θεωρησα , ολες τις δυνατες κινησεις που μπορει να κανει ο pacman απο αυτο το σημειο με την προυποθεση οτι δεν χτυπαει πανω σε τοιχο

-- Question 6

- Η Heuristic που υλοποιησα ουσιαστικα υπολογιζει για τον συγκριμενο current_node οπου βρισκεται τωρα την ΜΙΚΡΟΤΕΡΗ αποσταση απο ΟΛΕΣ τις γωνιες (corners) ΞΕΧΩΡΙΣΤΑ που ΔΕΝ εχουν επισκεφτει.
  και επιστρεφει το αθροισμα ολων των μικροτερων αποστασεων απο το current_node για ολες τις μη επισκεψιμες γωνιες.

-- Question 7

Στην FoodHeuristic αν δεν υπαρχει αλλο Remaining_Food επιστρεφει 0 . Αλλιως η Heuristic θα υπολογισει την αποσταση αναμεσα σε ολους τους κομβους που εχουν ακομα φαγητο και θα επιστρεψει την ΜΕΓΑΛΥΤΕΡΗ δυνατη αποσταση απο το current_node μεχρι το Food_Node.