Student Name: Κωνσταντίνος Φράγκος
Student ID: sdi2000207

Τεχνητή Νοημοσύνη
Readme

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

Ερώτημα 1:

Υλοποίηση του DFS χρησιμοποιόντας στίβα. Η λογική είναι η εξής:

- Βάζουμε το initial state στην στίβα και στη λίστα με τα επισκεπτόμενα states
- Βγάζουμε ένα στοιχείο από τη στίβα
- Αν είναι goal state τότε επιστρέφουμε το μονοπάτι
- Βάζουμε όλα τα παιδιά του state στη στίβα και τι λίστα εκτός από αυτά που έχουν ξαναμπεί παλαιότερα (Με τη χρήση ενός visitedlist στο οποίο αποθηκεύουμε τις ήδη επισκεπτόμενες καταστάσεις αυτό το ελέγχουμε)
- Επαναλαμβάνουμε από τη 2η κουκίδα μέχρι να αδειασει η στίβα

Αποθηκεύουμε μαζί με το state στη στίβα κάθε φορά και το μονοπάτι από το initial state στο εκάστοτε state
Θέλουμε να ψάξουμε πρώτα τα στοιχεία κατά βάθος και για αυτό χρησιμοποιούμε την στίβα καθώς το στοιχείο που θα βγάλουμε από τη στίβα, βάζοντας το παιδί του θα ψαχτεί αμέσως μετά.

Ερώτημα 2:

Η υλοποίηση είναι στην κυριολεξία copy paste της υλοποίησης του παραπάνω ερωτήματος με την διαφορά ότι αντί για στίβα χρησιμοποιούμε ουρά. Αυτό διότι θέλουμε να ψάξουμε κατά πλάτος επομένως θέλουμε και τα στοιχεία που μπαίνουν πρώτα στην ουρά να βγουν και πρώτα ώστε να ψαχτεί επίπεδο επίπεδο ο γράφος μας. Δεδομένου ότι όλα τα μονοπάτια έχουν κόστος 1 και ψάχνοντας λύση επίπεδο επίπεδο, ο αλγόριθμος αυτός θα μας επιστρέψει βέλτιστη λύση

Ερώτημα 3:

Ακολουθάμε παρόμοια λογική με τα παραπάνω ερωτήματα αλλά στην προκειμένη περίπτωση χρησιμοποιούμε priority queue με προταιρεότητα το συντομότερο κόστος του μονοπατιού από το initial state μέχρι τον εκάστοτε κόμβο + το κόστος του επομένου.
Αποθηκεύουμε το κόστος από το initial state μέχρι το εκάστοτε σε έναν πίνακα total coot έτσι, κάθε φορά θα βγαίνει από την ουρά ο κόμβος με το μικρότερο συνολικό κόστος, άρα θα βρούμε και μία βέλτιστη λύση

Ερώτημα 4:

Ίδια υλοποίηση με την UCS με την μικρή διαφορά ότι σαν προτεραιότητα παίρνουμε την ίδια με της UCS + την heuristic value του εκάστοτε state

Ερώτημα 5:

Σαν state θα έχουμε την θέση του pac man και το πόσες γωνίες έχουμε επισκευτεί.
Χρησιμοποιούμε ένα tuple με τις συντεταγμένες του pac man (x, y) και μία λίστα που περιέχει τις γωνίες που έχει επισκευτεί ο pacman
Η αρχική κατάσταση είναι ο pacman να βρίσκεται στην αρχική του θέση, και να μην έχουμε επισκευτεί καμία γωνία. Άρα η λίστα είναι άδεια αρχικά
Η τελική κατάσταση είναι ο pacman να έχει επισκευτεί όλες τις γωνίες. Άρα η λίστα έχει μέγεθος 4 καθώς περιέχει και τις 4 γωνίες
Όσον αφορά την successor function , επιστέφει τις νέες συντεταγμένες του pacman μετά από μία κίνησή του όχι πάνω σε κάποιο τοίχο, και αν αυτές οι συντεταγμένες είναι σε γωνιακή περιοχή με φαγητό, τότε επιστρέφει και την λίστα αλλαγμένη, προσθέτοντάς της την γωνία αυτή.

Ερώτημα 6:

η heuristic που υλοποίησα ακολουθεί την εξής λογική:
Προσθέτουμε την manhatan απόσταση του pac man από την συντομότερη γωνία, μετά την απόσταση αυτής της γωνίας με την επόμενη συντομότερη γωνία και πάει λέγοντας μέχρι να επισκευτούμε όλες τις γωνίες.
Είναι συνεπής και παραδεκτή ως συνεπής καθώς $h(n) - h(n') \leq 1 \leq c(c, a, c')$
Αφού ένα βήμα από έναν κόμβο n σε έναν γειτονικό n' η manhatan απόσταση θα μεταβληθεί το πολύ 1, αφού το κάθε βήμα του pacman στην πίστα σε manhatan απόσταση κοστίζει 1 και επιπλέον $h(goalstate) = 0$

Ερώτημα 7:

Στην προκυμένη heuristic βρίσκουμε τις πραγματικές αποστάσεις του pacman με όλες τις κουκίδες, και επιστρέφουμε την μεγαλύτερη.
Είναι συνεπής και παραδεκτή ως συνεπής αφού $h(n) - h(n') \leq 1 \leq c(c, a, c')$
Για τον ίδιο λόγο με το ερώτημα 6 απλά στην προκειμένη περίπτωση έχουμε πραγματική και όχι manhatan απόσταση

Ερώτημα 8:

Ορίζουμε σαν goal state του προβλήματος την περίπτωση που ο pacman μας βρίσκεται πάνω σε περιοχή που υπάρχει φαγητό.
Αναζητάμε άπληστα την πιο γρήγορη και τοπική λύση, από τη στιγμή που έχουμε ορίσει το παραπάνω goal state όλοι οι κόμβοι που περιέχουν κουκίδα θεωρούντε λύση. Τρέχοντας BFS θα βρούμε την πιο κοντινή. Επομένως καλούμε την ήδη υλοποιημένη BFS από μας στο 2ο ερώτημα, με την καινούρια όμως μοντελοποίηση των καταστάσεων του προβλήματός μας.