Τεχνητή Νοημοσύνη

Project 1 - Σχόλια για τον κώδικα του pacman

Κωνσταντίνος Πυθαρούλιος
Α.Μ.: 111520 1600142

Question 1:

Στο ερώτημα αυτό έχουμε μια τυπική υλοποίηση του DFS σε python. Αρχικά δημιουργούμε για να μας βοηθήσει μια λίστα Visited στην οποία θα αποθηκεύονται όσα nodes έχουμε ήδη επισκεφτεί ωστέ να μην τα επισκεφτούμε εκνέου αλλά και ένα dictionary με όνομα parentNode, που αντιστοιχίζει έναν κόμβο στον πρόγονό του.

Επομένως, ξεκινώντας από τον αρχικό κόμβο κατεβαίνουμε προς τα κάτω στους απογόνους. Όταν εντοπιστεί ένας κόμβος-στόχος, με βήματα προς τα πίσω βρίσκουμε ποιοι άλλοι κόμβοι μεσολάβησαν μέχρι αυτόν και ενσυνεχεία αντιστρέφουμε αυτή τη σειρά για να πάρουμε τελικά το μονομάτι από τον αρχικό κόμβο ως τον κόμβο στόχο.

Question 2

Εδώ κάναμε μια μικρή τροποποίηση σε σχέσε με τις διαφάνειες προκειμένου ο autograder να μας δώσει το 3/3. Συγκεκριμένα, φτιάξαμε τον αλγόριθμό μας έτσι ώστε να ελέγχει αν ένα node είναι κόμβος-στόχος **κατά την εξαγωγή** του από την ουρά και όχι κατά την εισαγωγή του σε αυτή.

Ελαφρώς παραλλαγμένος είναι επίσης ο τρόπος που υπολογίζουμε τα paths μέχρι από τον αρχικό κόμβο μέχρι τον κόμβο στόχο. Δεν υπέρχε κάποια ανάγκη για να γίνει αυτό. Απλά το δοκίμασα.

Question 3

Εδώ κάποια ιδιαίτερη παραδοχή δεν χρειάζεται να γίνει. Σε αντίθεση με τα προηγούμενα, χρησιμοποιούμε ουρά προτεραιότητας, με βάση τα costs.

Question 4

Ουσιαστικά, πρόκειται για την ίδια υλοποίηση με αυτή του uniformCostSearch με τη μόνη διαφορά ότι τώρα στο cost προστίθεται και το heuristic value.

Question 5

Για το κομμάτι αυτό δεν έχει γίνει κάποια συγκεκριμένη παραδοχή. Υλοποιήθηκε ό τι ακριβώς ανέφεραν τα υπάρχοντα σχόλια. Προστέθηκαν απλά κάποια νέα σχόλια, που να εξηγούν τι περίπου κάνουμε.

Question 6

Αρχικά, βρίσκουμε και τοποθετούμε σε μια λίστα όλα τα unvisited corners.
Στη συνέχεια βρίσκουμε την κοντινότερη γωνία(κόμβος-στόχος) και την αφερούμε από τη λίστα με τα unvisited corners.
Επαναλαμβάνουμε την ίδια διαδικασία για όλες τις γωνίες(κόμβους-στόχους) και παράλληλα αθροίζουμε τα κόστη που περιλαμβάνουν τα heuristic values.
Άρα τελικά, έχουμε ένα κάτω όριο του μονοπατιού που μπορεί να ακολουθήσει το pacman για να φτάσει σε όλες τις γωνίες.

Question 7

Η ιδέα για τον σχεδιασμό του αλγορίθμου αυτού του ερωτήματος προέκυψε από ένα σχόλιο του Antonio Juric στο stackoverflow *(https://stackoverflow.com/questions/9994913/pacman-what-kinds-of-heuristics-are-mainly-used)*.
Ουσιαστικά, η λύση χωρίζεται σε τρία βήματα:

1. Αρχικά βρίσκουμε την απόσταση μεταξύ των δύο φρούτων που βρίσκονται πιο μακριά μεταξύ τους. Έστω ότι την ονομάζουμε x.
2. Έπειτα, υπολογίζουμε την απόσταση από την τρέχουσα θέση του pacman, μέχρι το κοντινότερο από τα δύο φρούτα που βρήκαμε στο 1. Έστω ότι την ονομάζουμε y.
3. Επιστρέφουμε το άθροισμα των x,y.

Η συνάρτηση μπορεί μεν να αργεί αισθητά να εκτελεστεί, ωστόσο είναι αρκετά αποδοτική ως προς τα expanded nodes, οπότε και γενικά την θεωρούμε ικανοποιητική.

Question 8

Έχοντας δοκιμάσει διάφορες μεθόδους αναζήτησης από αυτές που υλοποιήσαμε, προτιμήσαμε για τα αποτελέσματά της, την bfs την οποία και καλούμε.