Read Me Για Τον Κώδικα Του Pac man

Άγγελος Τσιτσόλη sdi2000200

Νοέμβριος 13, 2022

Q1-Q4

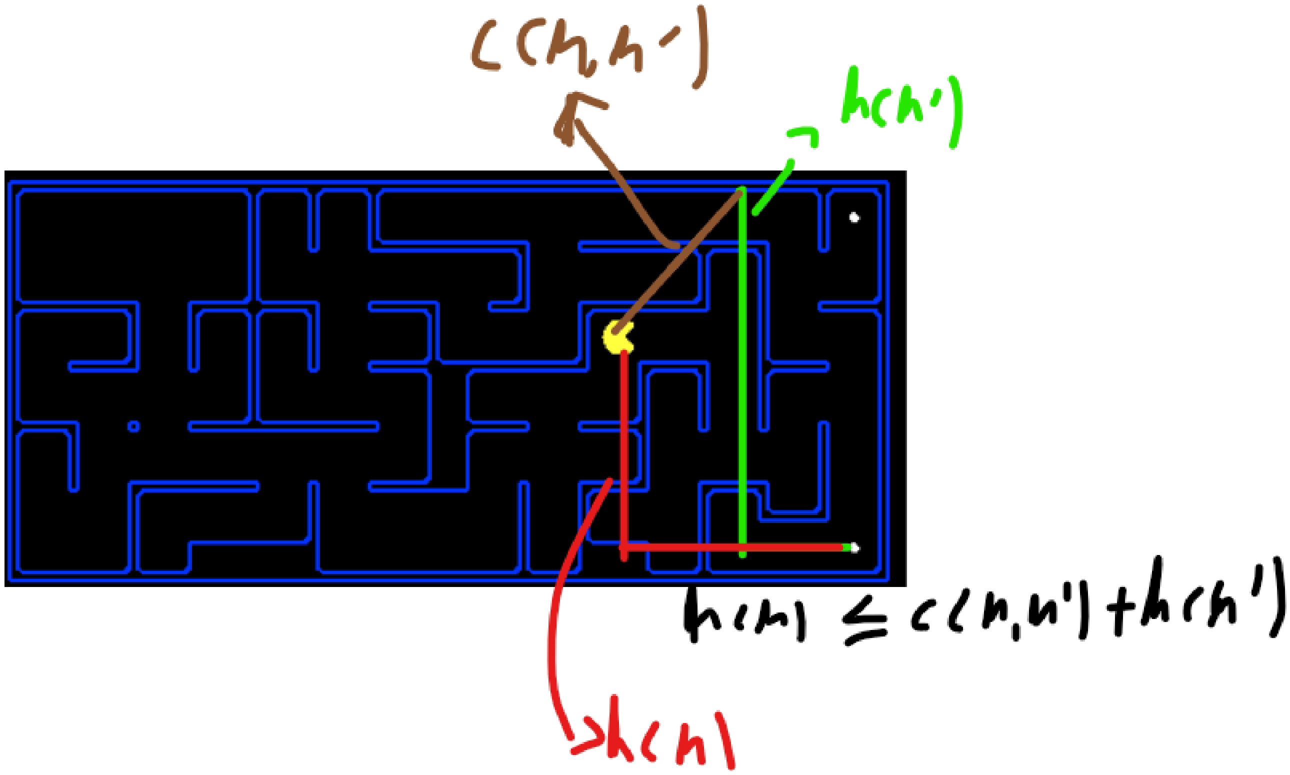
Για τα ερωτήματα αυτά ακολούθησα απλώς τον αλγόριθμο GraphSearch στην σελίδα 53 στις διαφάνειες της Ενότητας 2 .Συγκεκριμένα για το ερώτημα 1 που μας ζητείται να υλοποιήσουμε τον αλγόριθμο DFS , ακολούθησα τον αλγόριθμο των διαφανειών και ως σύνορο έβαλα μια στοίβα . Ομοίως για το ερώτημα 2 το οποίο ζητάει τον αλγόριθμο bfs χρησιμοποίησα ως σύνορο μια ουρά. Στην συνέχεια τα δύο επόμενα ερώτηματα(q3,q4) χρειάζονται και μερικές επιπρόσθετες αλλαγές.Δηλαδή το ερώτημα q3 μας ζητάει τον αλγόριθμο του UCS ο οποίος αλγόριθμος, προκειμένου να υλοποιηθεί απαιτεί για σύνορο μια σειρά προτεραιότητας . Η σειρά προτεραιότητας εκτός απο τον κόμβο που δέχεται απαιτεί να καθοριστεί και η προτεραιότητα του κόμβου σε σχέση με τους άλλους κόμβους που θα βρίσκονται στην ουρά προτεραιότητας . Στο ερώτημα q3 αυτή η προτεραιότητα για κάθε κόμβο θα είναι το κόστος που χρειάστηκε για να φτάσουμε σ'αυτό τον κόμβο κατά την εκτέλεση του αλγορίθμου UCS . Στο ερώτημα q4 χρησιμοποιείται ξανά ως σύνορο η δομή της ουράς προτεραιότητας απλώς στην περίπτωση αυτή η προτεραιότητα κάθε κόμβου που εισέρχεται στην ουρά προτεραιότητας θα είναι το άθροισμα του κόστους για να φτάσουμε μέχρι τον κόμβο αυτό συν του κόστους που μας παρέχει η ευρετική συνάρτηση.

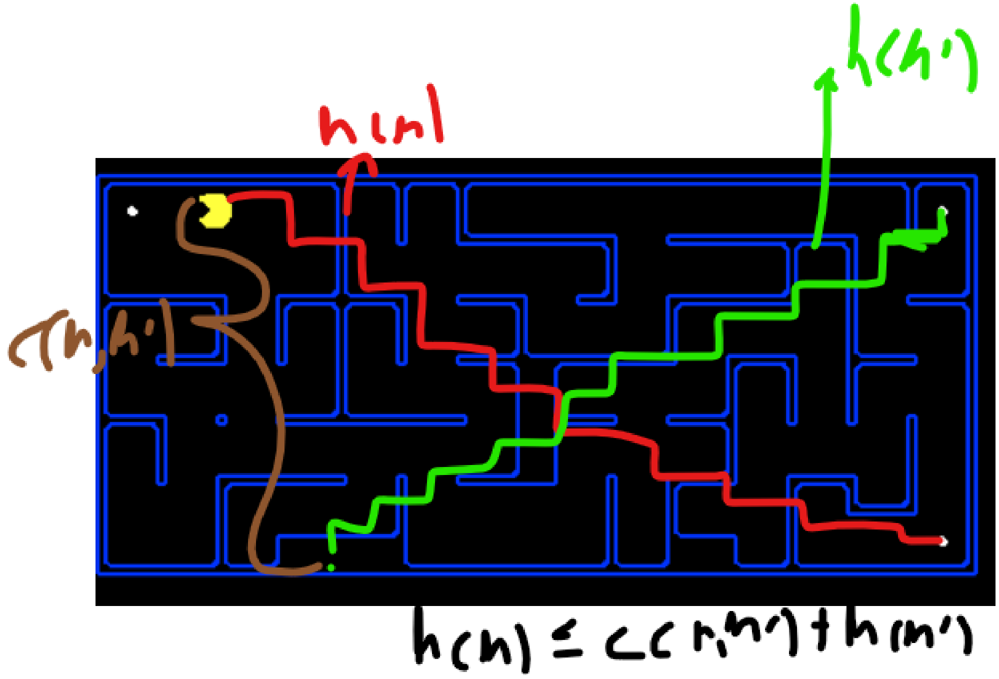
Q5

Το σκεπτικό για το ερώτημα αυτό ήταν το εξής : Ουσιαστικά κάθε κόμβος θα συμπεριλάμβανε και μια λίστα μετις γωνίες που πρέπει να επισκεφθούν απο το pacman και κάθε φορά που επισκέπτεται μία απο τις γωνίες τότε αφαιρείται η γωνία απο την λίστα. Δηλαδή ξεκινάμε με την αρχική κατάσταση η οποία θα διαθέτει μια λίστα με όλες τις γωνίες και κάθε φορά που το pacman παιρνάει απο μία γωνία θα αφαιρείται η γωνία απο την λίστα , συγκεκριμένα όταν λέω οτι θα αφαιρείται η γωνία, εννοώ πως ο επόμενος successor που δημιουργείται θα διαθέτει μια λίστα με τις γωνίες που έχει ακόμα το pacman να επισκεφθεί (οι γωνίες δηλαδή πλυν αυτήν που επισκεφθήκαμε). Καθώς μειώνονται οι γωνίες στις οποίες πρέπει περάσει το πακμαν κάποια στιγμή δεν θα υπάρχουν γωνίες να επισκεφθεί , επομ΄πενως αυτό σημαίνει ότι έφτασε σε κατάσταση στόχου. Η συνάρτηση isGoalState δηλαδή θα επιστρέψει true όταν ελέγξει κόμβο που περιέχει λίστα κενή απο γωνίες

Q6

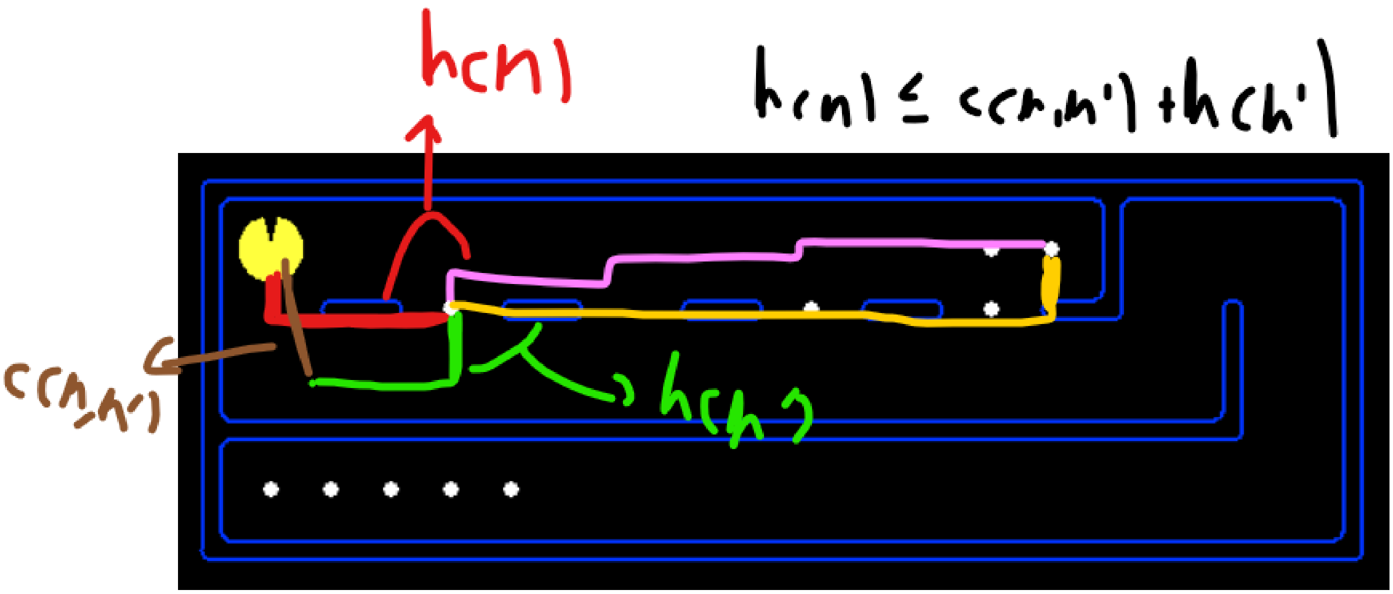
Στο ερώτημα αυτό χρειάζεται η δημιουργία μιας ευρετικής η οποία θα χρησιμοποιηθεί απο τον A* προκειμένου το pacman να περάσει και απο τις τέσσερις γωνίες , ώστε να φάει τις τελείες που βρίσκονται στις τοποθεσίες αυτές. Η τελική κατάσταση στο πρόβλημα αυτό θα είναι το pacman να έχει περάσει απο όλες τις γωνίες τρώγοντας αυτές τις τελείες που βρίσκονται εκεί.Ουσιαστικά το σκεπτικό για την δημιουργία της ευρετικής ήταν να βρώ μια συνάρτηση η οποία για μια κατάσταση θα μου λέει το κόστος μέχρι τον τελικό στόχο.Επομένως για την υλοποίηση της ευρετικής αυτής σκέφτηκα ένα πιο χαλαρωμένο πρόβλημα χωρίς τοίχους και πολύ πιο απλό απο το αρχικό . Ένα τέτοιο πρόβλημα θα ήταν ένα το οποίο θα είχε ως στόχο να φτάσουμε απο μια κατάσταση στην κοντινότερη γωνία είτε στην μακρύτερη γωνία .Η ευρετική συνάρτηση ουσιαστικά σαυτήν την περίπτωση (στο μικρότερο αυτό πρόβλημα) θα κάνει ακριβώς ότι και η βέλτιστη λύση δηλαδή θα βρίσκει το κόστος της μικρότερης ή της μεγαλύτερης απόστασης απο την κατάσταση μας μέχρι μια γωνία δηλαδή .Άρα θα βρίσκει το βέλτιστο κόστος γιατί κάνει ακριβώς ότι και η βέλτιστη λύση για κάθε κατάσταση . Ωστόσο επειδή το κόστος της ευρετικής που βρίσκει την γωνία που είναι σε μεγαλύτερη απόσταση από την κατάσταση που εξετάζουμε θα είναι σίγουρα μεγαλύτερο απο το κόστος της απόστασης της κατάστασης που εξετάζουμε απο την κοντινότερη γωνία και προφανώς πάντα με κόστος μικρότερο ή ίσο απο την βέλτιστη διαδρομή τότε η ευρετική που βρίσκει την μεγαλύτερη απόσταση θα είναι καλύτερη για το αρχικό μας πρόβλημα διότι το κόστος διαδρομής που βγάζει θα είναι πιο κοντά στο βέλτιστο κόστος διαδρομής απο ότι η ευρετική που που υπολογίζει το την μικρότερη απόσταση απο κατάσταση.Άρα η ευρετική μας εφόσον παράγει κόστος που προσεγγίζει κατά πολύ το κόστος της βέλτιστης διαδρομής στο απλό πρόβλημα άρα στο μεγαλύτερο και πιο γενικό μας πρόβλημα θα είναι παραδεκτή .Η ευρετική προφανώς δεν θα υπερεκτιμά το πραγματικό κόστος το οποίο φανερώνει ότι θα είναι παραδεκτή.Όσον αφορά την συνέπεια ισχύει ότι εφόσον τελική μας κατάσταση είναι να περάσει απο όλες τις γωνίες όταν φτάσει κακι στην τελευταία γωνία η ευρετική θα βγάλει κόστος h(last-corner)=0 άρα επιβεβαιώνεται η μία προυπόθεση συνέπειας . Έπειτα θα ισχύει ο τύπος h(n)$\leq c(n,n') + h(n')$ όπου h(χ) το κόστος μιας κατάστασης που δίνει η ευρετική, n μια κατάσταση , n' μια γειτονική κατάσταση και c(n,n') το πραγματικό κόστος μεταξύ των καταστάσεων n και n'. Έστω΄μερικές περιπτώσεις : Για την περίπτωση όπου φτάσαμε στην τελευταία τελεία που βρίσκεται σε γωνία ενώ υπάρχουν και άλλες τελείες ακόμα για επίσκεψη τότε θα έχουμε ότι : h(n)$\leq c(n,n') + h(n')$ <=> 0$\leq c(n,n') + h(n)$ εφόσον δεν υπάεχουν άλλες γωνίς το κόστος της ευρετικής για την n είναι ίσο με μηδέν και το κόστος της ευρετικής μιας γειτονικής θα είναι ίσο με την απόσταση τους εφόσον μόνο αυτή η γωνία απέμεινε άρα 0$\leq 2 * c(n,n')$. Σε περίπτωση που φτάνουμε σε κατάστση που είναι γωνία και υπάρχουν ακόμα τρείς γωνίες που δεν έχουν επισκεφθεί (για παράδειγμα είμαστε στην κάτω αριστερή γωνία του ορθογωνίου που βγαίνει όταν τρέχουμε το ερώτημα αυτό στο τερμιναλ)τότε έχουμε ότι η πάνω δεξιά γωνία είναι η μακρυνότερη για την κατάσταση μας σύμφωνα με την ευρετική , έπειτα απο μία γειτονική κατάσταση το κόστος για την μακρυνότερη γωνία που θα είναι προφανώς η ίδια και για την γειτονική γειτονική θα είναι ενδεχομέμενως λίγο μικρότερο απο το κόστος που δίνει η ευρετική για την κατάσταση μας (που είναι η γωνία) άρα προσθέτωντας στο κόστος που δίνει η ευρετική για την γειτονική το κόστος της πραγματικής απόστασης των δύο γειτονικών καταστάσεων τότε προφανώς θα ισχύει ο τύπος συνέπειας.Πολλές περιπτώσεις προκύπτουν όπως η εξής παρακάτω εικόνα κατά την οποία το h(n) και h(n') υπολογίζονται στον κώδικα μας απο την ευρετική manhattan distance δηλαδή δεν παίρνονται υπόψην καθόλου οι τοίχοι ως εμπόδια.Όπως φαίνεται στην εικόνα ξεκάθαρα ισχύει ο τύπος συνέπειας καθώς το κόστος της κόκκινης γραμμής που αποτελεί την απόσταση της κατάστασης μας μέχρι την μακρυνότερη γωνία είναι μικρότερο απο το κόστος . Όπως φαίνεται και στα άλλα παραδείγματα στις εικόνες .:

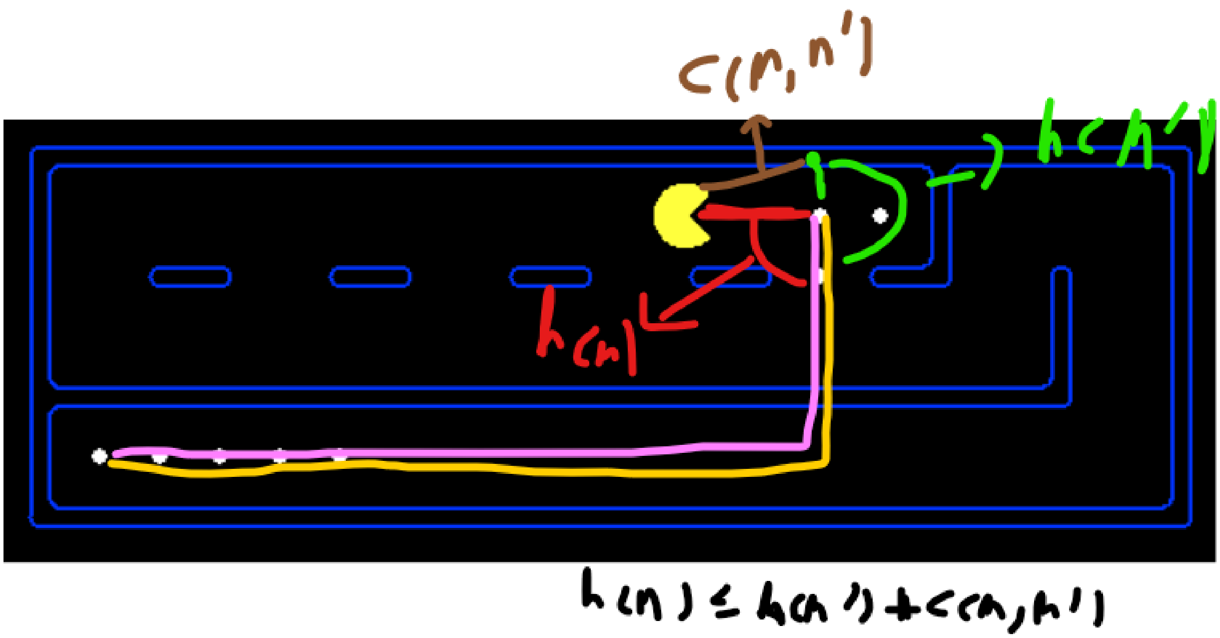




Q7

Σαυτό το πρόβλημα μας ζητείται να βρούμε ένα path προκειμένου το pacman να καταφέρει να φάει όλες τις τελείες. Συγκεκριμένα πρέπει να βρούμε μια ευρετική μέσω της οποίας ο αλγόριθμος A* θα βρίσκει όλες τις τελείες και το πακμαν θα τις τρώει. Προκειμένου να βρώ μια ευρετική ακολούθησα τον γνωστό συλλογισμό, δηλαδή να προσπαθήσω να λύσω ένα πιο χαλαρωμένο πρόβλημα απο αυτό που έχουμε να λύσουμε , συγκεκριμένα να βρώ τη λύση σε ένα πιο απλό πρόβλημα τέτοιου τύπου.Ουσιαστικά ένα πιο απλό πρόβλημα απο αυτό που μας ζητείται θα ήταν ένα πρόβλημα με τελικό στόχο το πακμαν απο την κατάσταση που βρίσκεται κάθε φορά να φάει την πιο μακρινή τελεία μεταξύ των τελείων που βρίσκονται στις γωνίες που έχουν μείνει και δεν έχουν φαγωθεί ενώ έχει ήδη φάει μία τελεία που δεν βρίσκεται σε γωνία. Στο πιο απλό αυτό πρόβλημα που περιέγραψα η βέλτιστη διαδρομή που μπορεί να ακολουθήσει το πακμαν ώστε να φτάσει στον τελικό στόχο και να μην ελέγξει περισσότερες απο όσες πρέπει καταστάσεις , θα είναι αυτή στην οποία θα επιλέξει την κοντινότερη τελεία σαυτό και έπειτα απο εκεί θα επιλέξει την μακρυνότερη τελεία απο αυτές των γωνίων . Η ευρετική που θα μπορούσαμε να χρησιμοποιήσουμε για το απλό αυτό πρόβλημα είναι η εξής : απο μια κατάσταση που είμαστε , να βρούμε το κόστος της απόστασης απο το πακμαν μέχρι την κοντινότερη τελεία που θα έχει και στην συνέχεια να βρούμε απο την θέση που βρίσκεται η πιο κοντινή τελεία που βρήκαμε ότι θα πάει ο πακμαν , τις αποστάσεις μέχρι τις τελείες που βρίσκονται στις γωνίες , να τις συγκρίνουμε (τις αποστάσεις μεταξύ κοντινής τελείας που τρώει το πακ μαν και κάθε μιας γωνίας στις οποίες βρίσκονται τελείες) , να κρατήσουμε την μεγαλύτερη απόσταση και στην συνέχεια να προσθέσουμε το κόστος απο την απόσταση του πακμαν μέχρι την πιο κοντινή τελεία με το κόστος απο την θέση αυτή μέχρι την πιο μακρινή τελεία που βρίσκεται σε γωνία. Η ευρετική μας επομένως θα είναι παραδεκτή, διότι δεν θα υπερεκτιμά το πραγματικό κόστος της πραγματικής διαδρομής που θα κάνει το πακμαν , αυτό φαίνεται απο το γεγονός ότι η ευρετική θεωρητικά θα βρεί έτσι το κόστος της βέλτιστης διαδρομής, ωστόσο στην πράξη δεν ξέρουμε αν ο A* χρησιμοποιώντας την ευρετική μας θα λύσει το πρόβλημα μέσω της βέλτιστης λύσης , αλλά ξέρουμε σίγουρα ότι δεν θα υπερεκτιμά το πραγματικό κόστος.Το κόστος δηλαδή της διαδρομής που θα βρεί η ευρετική θα είναι είτε ίσο είτε μικρότερο απο το πραγματικό κόστος της ευρετικής συνάρτησης μας . Επομένως η ευρετική που βρήκαμε στο απλό πρόβλημα θα θεωρήσουμε ότι προσεγγίζει κατά πάρα πολύ την βέλτιστη λύση . Άρα εφόσον ισχύει ότι η βέλτιστη λύση σε ένα απλό πρόβλημα αποτελεί μια παραδεχτή ευρετική συνάρτηση στο αρχικό πρόβλημα .Με λίγα λόγια η ευρετική που βρήκαμε για το απλό ερώτημα μπορεί να χρησιμοποιηθεί ως ευρετική στο αρχικό πρόβλημα να φάει όλες τις κουκίδες δηλαδή το πακμαν και μάλιστα εφόσον είναι τόσο κοντά σην βέλτιστη διαδρομή για τη λύση του απλού προβλήματος που θεωρούμε και ότι η ευρετική θα βρεί την βέλτιστη λύση και άρα θα είναι παραδεχτεί για το αρχικό μας πρόβλημα.Χρησιμοποιώντας αυτην την πληροφορία που δίνει η ευρετική απο τον A* δεν θα ελεγχθούν τόσοι κόμβοι όσο χωρίς αυτήν και θα βρεθεί η λύση γρηγορότερα εφόσον μειώνονται οι έλεγχοι.Όσον αφορά την συνέπεια του προβλήματος θα ισχύει η ανισότητα h(n)≤ $c(n, n') + h(n')$ όπου h(χ) το κόστος μιας κατάστασης που δίνει η ευρετική, n μια κατάσταση , n' μια γειτονική κατάσταση και c(n,n') το πραγματικό κόστος μεταξύ των καταστάσεων n και n' επίσης ισχύει ότι h(goal-state)=0. Ο τύπος στην περίπτωση μας όπου κάθε ενέργεια ισοδυναμεί με 1 θα είναι κάπως έτσι h(n)≤ $1 + h(n')$ επομένως η διαφορά h(n)-h(n') δεν θα είναι μεγαλύτερη απο ένα . Έστω μερικές περιπτώσεις οι εξής :(με κόκκινη γραμμή για την κατάσταση που βρίσκεται το πακμαν και το κόστος της ευρετικής του υπολογίζεται απο την ροζ και την κόκκινη ομοίως και για μια κατάσταση πράσινη το κίτρινο και το πράσινο αποτελούν το κόστος που παράγει η ευρετική για αυτήν)





Q8

Αρχικά στο ερώτημα αυτό μας ζητείται να προσθέσουμε έναν αλγόριθμο που θα βρίσκει το μονοπάτι προς την κοντινότερη τελεία .Ο αλγόριθμος που χρησιμοποίησα είναι ο bfs, καθώς σύμφωνα με τις διαφάνειες είναι πλήρης αλγόριθμος δηλαδή βρίσκει πάντα λύση στο πρόβλημα μας και είναι και βέλτιστος διότι στην περίπτωση μας έχουμε ίδιο κόστος όσον αφορά τις ενέργειες , όλες οι ενέργειες έχουν κόστος 1 επομένως θα είναι και βέλτιστος . Μας ζητείται επίσης να προσθέσουμε και μια προυπόθεση για τον στόχο . Με λίγα λόγια αν μια κάτάσταση αποτελεί κατάσταση στην οποία υπάρχει φαγητό για το πακ μαν τότε αυτό αποτελεί κατάσταση στόχου . Άρα ελέγχουμε αν υπάρχει η κατάσταση που εξετάζουμε εκείνη την στιγμή στην λίστα με τις καταστάσεις φαγητών για τις τελείες που θα φάει το πακ μαν. Στην συνέχεια ακολουθούν ορισμένα σχόλια όσον αφορά την εκφώνηση η οποία κάνει λόγο για άπληστους αλγορίθμους και ότι δεν βρίσκουν πάντοτε το βέλτιστο μονοπάτι.Ενας άπληστος αλγόριθμος δεν μπορεί να λειτουργεί πάντα , δηλαδή στην περίπτωση μας δεν σημαίνει ότι ένας αλγόριθμος που επιλέγει συνεχώς την τελεία που έχει τη μικρότερη απόσταση απο την κατάσταση μας θα βρεί και το βέλτιστο μονοπάτι .Ένας άπληστος αλγόριθμος θα βρεί τη λύση αλλά δεν μπορούμε να πούμε με σιγουριά αν θα βρεί τη βέλτιστη πάντα. Εδώ μπορεί να οδηγήσει το πακμαν να φάει την πιο κοντινή τελεία εκείνη την στιγμή και να συνεχίσει απο εκεί να τρώει αλλά έπειτα να απορρίψει μια πιο μακρυνή τελεία η οποία εν τέλει να το οδηγούσε στο βέλτιστο μονοπάτι. Δηλαδή ένας άπληστος αλγόριθμος καθώς αποφασίζει κάθε φορά εκείνη την στιγμή με ένα μόνο κριτήριο το ποια τελεία είναι πιο κοντά και δεν κάνει αποφάσεις για να συμβάλλει μελλοντικά στον βέλτιστο τρόπο άφιξης στο τελικό αποτέλεσμα , 'χάνει' κάποιο μονοπάτι που σε κάποια σημεία μπορεί να παραβιάζει τα κριτήρια του αλγορίθμου, αλλά εν τέλει αποτελεί την βέλτιστη λύση και συνολικά έχει εν τέλει το μικρότερο κόστος.