

1η Σειρά Θεωρητικών Ασκήσεων

Τεχνητή Νοημοσύνη

Κωνσταντίνος Κασφίκης

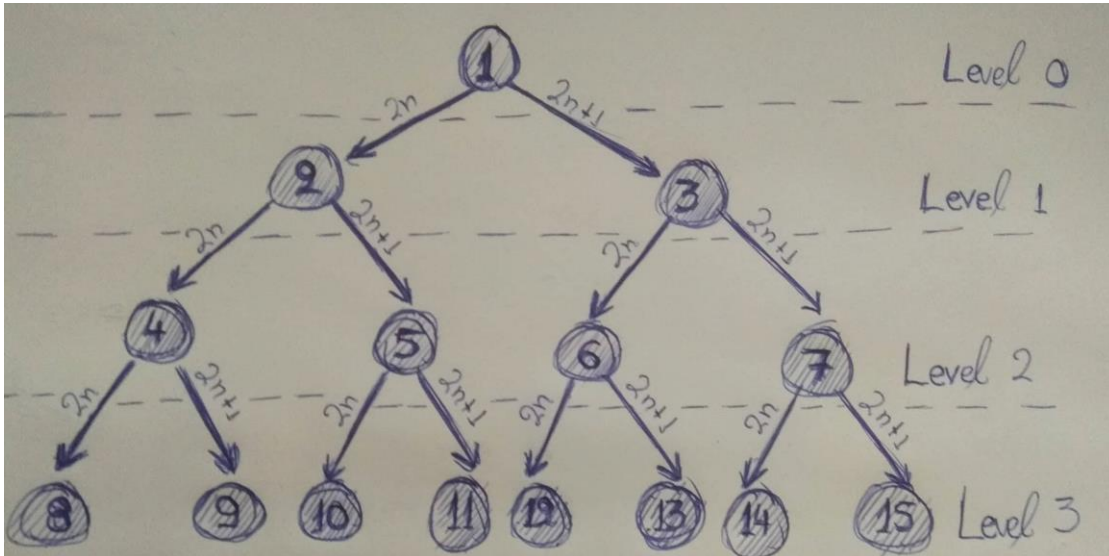
2013030108

Άσκηση 1η

	Μέτρο Απόδοσης	Περιβάλλον	Ενέργειες	Αισθητήρες
Ρομποτικός Καλαθοσφαιριστής	Κατοχή Μπάλας Ποσοτό εύστοχων Βολών	Γήπεδο μπάσκετ Υπόλοιποι Παίκτες Το περιβάλλον είναι δυναμικό λόγων των υπόλοιπων παικτών	Κίνηση στο γήπεδο Κινήσεις που αφορούν πάσες,σκάσιμο μπάλας,σουτ	Αισθητήρες Απόστασης ή και κάμερα ώστε να μπορεί ο πράκτορας να ξέρει την θέση του στο γήπεδο καθώς και την θέση των συμπαικτών αντιπάλων.
Διαχωριστής Μήλων – Πορτοκαλιών	Ποσοστό μήλων έναντι πορτοκαλιών Ποσοστό λάθους	Βιομηχανικό εργοστάσιο Στατικό Περιβάλλον	Συλλογή φρούτου Τοποθέτηση φρούτου στο κατάλληλο καλάθι	Κάμερα , ώστε να μπορεί να ξεχωρίζει τα μήλα από τα πορτοκάλια με την βοήθεια κατάλληλου αλγόριθμου
Χρηματιστηριακός Επενδυτής	Ποσοστό Κέρδους από αγοραπωλησίες μετοχών	Online Πλατφόρμα για χρηματιστηριακές συναλλαγές Δυναμικό Περιβάλλον	Αγορά μετοχών Πώληση μετοχών	Εφόσον ο χρηματιστηριακός επενδυτής λειτουργεί σε virtual περιβάλλον ,χωρίς την ανάγκη σύνδεσης με τον πραγματικό κόσμο δεν χρησιμοποιεί αισθητήρες

Άσκηση 2^η

A) Με δεδομένο ότι η αρχική κατάσταση είναι 1 και η συνάρτηση επόμενων καταστάσεων επιστρέφει 2 καταστάσεις ($2n$ και $2n+1$) μπορούμε να δημιουργήσουμε δέντρο καταστάσεων το οποίο εκτείνεται μέχρι την απαιτούμενη τιμή 15.



B) Έχοντας ως στόχο την κατάσταση 11 και ως κατάσταση εκκίνησης την κατάσταση 1

1) Breadth First Search

Αποτελεί έναν αλγόριθμο αναζήτησης ο οποίος διασχίζει το δέντρο καταστάσεων επίπεδο προς επίπεδο.

Για την υλοποίηση του αλγόριθμου χρειαζόμαστε μια λίστα όπου θα αποθηκεύονται οι καταστάσεις τις οποίες έχουμε ήδη επισκεφτεί (V) και μια FIFO (First In - First out) ουρά η οποία περιέχει τα στοιχεία προς επεξεργασία (Q). Επεξεργαζόμαστε τις καταστάσεις μία προς μία, προσθέτοντας την εκάστοτε κατάσταση (current) στην λίστα Visited (V) και προσθέτοντας τις καταστάσεις-παιδιά της current στην ουρά Q. Η αναζήτηση σταματά όταν η κατάσταση current είναι η κατάσταση-στόχος.

Βήματα που ακολουθεί ο αλγόριθμος BFS:

Κατά την έναρξη του αλγορίθμου τόσο η λίστα V όσο και η ουρά Q είναι κενές.

Το πρώτο στοιχείο που επεξεργάζεται είναι η κατάσταση εκκίνησης δηλαδή η κατάσταση 1

$V = \{1\}$

$Q = \{1\}$

Output = {}

Αφαιρούμε από την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 1 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 1, δηλαδή τις καταστάσεις 2 και 3.

$V = \{1, 2, 3\}$

$Q = \{2, 3\}$

Output = {1}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 2 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 2, δηλαδή τις καταστάσεις 4 και 5.

$V = \{1,2,3,4,5\}$

$Q = \{3,4,5\}$

Output = {1,2}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 3 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 3, δηλαδή τις καταστάσεις 6 και 7.

$V = \{1,2,3,4,5,6,7\}$

$Q = \{4,5,6,7\}$

Output = {1,2,3}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 4 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 4, δηλαδή τις καταστάσεις 8 και 9.

$V = \{1,2,3,4,5,6,7,8,9\}$

$Q = \{5,6,7,8,9\}$

Output = {1,2,3,4}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 5 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 5 , δηλαδή τις καταστάσεις 10 και 11.

$V = \{1,2,3,4,5,6,7,8,9,10,11\}$

$Q = \{6,7,8,9,10,11\}$

Output = {1,2,3,4,5}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 6 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 6, δηλαδή τις καταστάσεις 12 και 13.

$V = \{1,2,3,4,5,6,7,8,9,10,11,12,13\}$

$Q = \{7,8,9,10,11,12,13\}$

Output = {1,2,3,4,5,6}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 7 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 7, δηλαδή τις καταστάσεις 14 και 15.

$V = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}$

$Q = \{8,9,10,11,12,13,14,15\}$

Output = {1,2,3,4,5,6,7}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 8, η οποία δεν έχει καταστάσεις-παιδιά άρα δεν προστίθεται κάποια κατάσταση στην ουρά.

$V = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}$

$Q = \{9,10,11,12,13,14,15\}$

Output = {1,2,3,4,5,6,7,8}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 9, η οποία δεν έχει καταστάσεις-παιδιά άρα δεν προστίθεται κάποια κατάσταση στην ουρά.

V = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}

Q = {10,11,12,13,14,15}

Output = {1,2,3,4,5,6,7,8,9}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 10, η οποία δεν έχει καταστάσεις-παιδιά άρα δεν προστίθεται κάποια κατάσταση στην ουρά.

V = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}

Q = {11,12,13,14,15}

Output = {1,2,3,4,5,6,7,8,9,10}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 11. Η κατάσταση 11 είναι ο στόχος της αναζήτησης, συνεπώς η αναζήτηση τερματίζεται.

V = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}

Q = {12,13,14,15}

Output = {1,2,3,4,5,6,7,8,9,10,11}

2) Depth-Limited με όριο 3

Ο Depth first search αποτελεί έναν αλγόριθμο αναζήτησης ο οποίος εμβαθύνει στο δέντρο καταστάσεων όσο κάτι τέτοιο είναι εφικτό, αλλιώς προβαίνει σε backtrack. Με τον όρο backtrack εννοείται ότι εφόσον δεν υπάρχουν περισσότεροι κόμβοι κατα μήκος της τρέχουσας διαδρομής, ο αλγόριθμος ακολουθεί την διαδρομή αυτή με αντίθετη φορά (προς τα πίσω) με σκοπό την εύρεση καταστάσεων που δεν έχουν επισκεφθεί και τις οποίες ο αλγόριθμος θα επισκεφθεί και θα εμβαθύνει κατά μήκος. Ο Depth-limited search αλγόριθμος είναι μια παραλλαγή του Depth-First-Search ο οποίος εισάγει ένα όριο (limit) που αντιστοιχεί στο μέγιστο αριθμό επιπέδων που επιτρέπονται στον αλγόριθμο να εμβαθύνει.

Για την υλοποίηση του αλγόριθμου χρειαζόμαστε μια λίστα όπου θα αποθηκεύονται οι καταστάσεις τις οποίες έχουμε ήδη επισκεφθεί (V) και μια LIFO (Last In - First out) ουρά η οποία περιέχει τα στοιχεία προς επεξεργασία (Q). Επεξεργαζόμαστε τις καταστάσεις μία προς μία, προσθέτοντας την εκάστοτε κατάσταση (current) στην λίστα Visited (V) και προσθέτοντας τις καταστάσεις-παιδιά της current στην ουρά Q. Η αναζήτηση σταματά όταν η κατάσταση current είναι η κατάσταση-στόχος.

Βήματα που ακολουθεί ο αλγόριθμος Depth-Limited (με όριο 3):

Κατά την έναρξη του αλγορίθμου τόσο η λίστα V όσο και η ουρά Q είναι κενές.

Το πρώτο στοιχείο που επεξεργάζεται είναι η κατάσταση εκκίνησης δηλαδή η κατάσταση 1

V = {1}

Q = {1}

Output = {}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 1 και προσθέτουμε στην

ουρά τις καταστάσεις - παιδιά της 1 , δηλαδή τις καταστάσεις 2 και 3.

$$V = \{1,2,3\}$$

$$Q = \{2,3\}$$

$$\text{Output} = \{1\}$$

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 2 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 2, δηλαδή τις καταστάσεις 2 και 3.

$$V = \{1,2,3,4,5\}$$

$$Q = \{4,5,3\}$$

$$\text{Output} = \{1,2\}$$

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 4 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 4, δηλαδή τις καταστάσεις 8 και 9.

$$V = \{1,2,3,4,5,8,9\}$$

$$Q = \{8,9,5,3\}$$

$$\text{Output} = \{1,2,4\}$$

Εφόσον το πρώτο στοιχείο της ουράς δηλαδή η κατάσταση 8 βρίσκεται στο επίπεδο 3, δεν εμβαθύνουμε άλλο. Αφαιρούμε την κατάσταση 8 απο την ουρά

$$V = \{1,2,3,4,5,8,9\}$$

$$Q = \{9,5,3\}$$

$$\text{Output} = \{1,2,4,8\}$$

Εφόσον το πρώτο στοιχείο της ουράς δηλαδή η κατάσταση 9 βρίσκεται στο επίπεδο 3, δεν εμβαθύνουμε άλλο. Αφαιρούμε την κατάσταση 9 απο την ουρά

$$V = \{1,2,3,4,5,8,9\}$$

$$Q = \{5,3\}$$

$$\text{Output} = \{1,2,4,8,9\}$$

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 5 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 5 , δηλαδή τις καταστάσεις 10 και 11.

$$V = \{1,2,3,4,5,8,9,10,11\}$$

$$Q = \{10,11,3\}$$

$$\text{Output} = \{1,2,4,8,9,5\}$$

Εφόσον το πρώτο στοιχείο της ουράς δηλαδή η κατάσταση 10 βρίσκεται στο επίπεδο 3, δεν εμβαθύνουμε άλλο. Αφαιρούμε την κατάσταση 10 απο την ουρά

$$V = \{1,2,3,4,5,8,9,10,11\}$$

$$Q = \{11,3\}$$

$$\text{Output} = \{1,2,4,8,9,5,10\}$$

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 11. Η κατάσταση 11 είναι ο

στόχος της αναζήτησης, συνεπώς η αναζήτηση τερματίζεται.

$V = \{1,2,3,4,5,8,9,10,11\}$

$Q = \{3\}$

$Output = \{1,2,4,8,9,5,10,11\}$

3) Iterative Deepening

Πραγματοποιούμε πολλαπλά Depth-Limited searches στο ίδιο γραφο καταστάσεων, αυξάνοντας απο αναζήτηση σε αναζήτηση το limit μέχρι να βρεθεί η κατάσταση στόχος.

Βήματα που ακολουθεί ο αλγόριθμος Iterative Deepening:

Για $limit = 0$

Το πρώτο στοιχείο που επεξεργάζεται είναι η κατάσταση εκκίνησης δηλαδή η κατάσταση 1

$V = \{1\}$

$Q = \{1\}$

$Output = \{\}$

Εφόσον το πρώτο στοιχείο της ουράς δηλαδή η κατάσταση 1 βρίσκεται στο επίπεδο 0, δεν εμβαθύνουμε άλλο. Αφαιρούμε την κατάσταση 1 απο την ουρά

$V = \{1\}$

$Q = \{\}$

$Output = \{1\}$

Έχοντας επισκεφτεί όλες τις πιθανές καταστάσεις για $limit = 0$ και εφόσον στο output δεν περιέχεται η κατάσταση στόχος, αυξάνουμε το limit κατα 1 και επαναλαμβάνουμε την διαδικασία

Για $limit = 1$

Το πρώτο στοιχείο που επεξεργάζεται είναι η κατάσταση εκκίνησης δηλαδή η κατάσταση 1

$V = \{1\}$

$Q = \{1\}$

$Output = \{\}$

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 1 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 1 , δηλαδή τις καταστάσεις 2 και 3.

$V = \{1,2,3\}$

$Q = \{2,3\}$

$Output = \{1\}$

Εφόσον το πρώτο στοιχείο της ουράς δηλαδή η κατάσταση 2 βρίσκεται στο επίπεδο 1, δεν εμβαθύνουμε άλλο. Αφαιρούμε την κατάσταση 2 απο την ουρά

$V = \{1,2,3\}$

$Q = \{3\}$

Output = {1,2}

Εφόσον το πρώτο στοιχείο της ουράς δηλαδή η κατάσταση 3 βρίσκεται στο επίπεδο 1, δεν εμβαθύνουμε άλλο. Αφαιρούμε την κατάσταση 3 απο την ουρά

$V = \{1,2,3\}$

$Q = \{\}$

Output = {1,2,3}

Έχοντας επισκεφτεί όλες τις πιθανές καταστάσεις για $limit = 1$ και εφόσον στο output δεν περιέχεται η κατάσταση στόχος, αυξάνουμε το $limit$ κατα 1 και επαναλαμβάνουμε την διαδικασία

Για $limit = 2$

Το πρώτο στοιχείο που επεξεργάζεται είναι η κατάσταση εκκίνησης δηλαδή η κατάσταση 1

$V = \{1\}$

$Q = \{1\}$

Output = {}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 1 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 1 , δηλαδή τις καταστάσεις 2 και 3.

$V = \{1,2,3\}$

$Q = \{2,3\}$

Output = {1}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 2 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 2, δηλαδή τις καταστάσεις 4 και 5.

$V = \{1,2,3,4,5\}$

$Q = \{4,5,3\}$

Output = {1,2}

Εφόσον το πρώτο στοιχείο της ουράς δηλαδή η κατάσταση 4 βρίσκεται στο επίπεδο 2, δεν εμβαθύνουμε άλλο. Αφαιρούμε την κατάσταση 4 απο την ουρά

$V = \{1,2,3,4,5\}$

$Q = \{5,3\}$

Output = {1,2,4}

Εφόσον το πρώτο στοιχείο της ουράς δηλαδή η κατάσταση 5 βρίσκεται στο επίπεδο 2, δεν εμβαθύνουμε άλλο. Αφαιρούμε την κατάσταση 5 απο την ουρά

$V = \{1,2,3,4,5\}$

$Q = \{3\}$

Output = {1,2,4,5}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 3 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 3 , δηλαδή τις καταστάσεις 6 και 7.

$V = \{1,2,3,4,5,6,7\}$

$Q = \{6,7\}$

Output = {12,4,5,3}

Εφόσον το πρώτο στοιχείο της ουράς δηλαδή η κατάσταση 6 βρίσκεται στο επίπεδο 2, δεν εμβαθύνουμε άλλο. Αφαιρούμε την κατάσταση 6 απο την ουρά

$V = \{1,2,3,4,5,6,7\}$

$Q = \{7\}$

Output = {1,2,4,5,3,6}

Εφόσον το πρώτο στοιχείο της ουράς δηλαδή η κατάσταση 6 βρίσκεται στο επίπεδο 2, δεν εμβαθύνουμε άλλο. Αφαιρούμε την κατάσταση 6 απο την ουρά

$V = \{1,2,3,4,5,6,7\}$

$Q = \{\}$

Output = {1,2,4,5,3,6,7}

Έχοντας επισκεφτεί όλες τις πιθανές καταστάσεις για $limit = 2$ και εφόσον στο output δεν περιέχεται η κατάσταση στόχος, αυξάνουμε το $limit$ κατα 1 και επαναλαμβάνουμε την διαδικασία

Για $limit = 3$

Το πρώτο στοιχείο που επεξεργάζεται είναι η κατάσταση εκκίνησης δηλαδή η κατάσταση 1

$V = \{1\}$

$Q = \{1\}$

Output = {}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 1 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 1 , δηλαδή τις καταστάσεις 2 και 3.

$V = \{1,2,3\}$

$Q = \{2,3\}$

Output = {1}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 2 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 2 , δηλαδή τις καταστάσεις 2 και 3.

$V = \{1,2,3,4,5\}$

$Q = \{4,5,3\}$

Output = {1,2}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 4 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 4 , δηλαδή τις καταστάσεις 8 και 9.

$V = \{1,2,3,4,5,8,9\}$

$Q = \{8,9,5,3\}$

Output = {1,2,4}

Εφόσον το πρώτο στοιχείο της ουράς δηλαδή η κατάσταση 8 βρίσκεται στο επίπεδο 3, δεν εμβαθύνουμε άλλο. Αφαιρούμε την κατάσταση 8 απο την ουρά

$V = \{1,2,3,4,5,8,9\}$

$Q = \{9,5,3\}$

Output = {1,2,4,8}

Εφόσον το πρώτο στοιχείο της ουράς δηλαδή η κατάσταση 9 βρίσκεται στο επίπεδο 3, δεν εμβαθύνουμε άλλο. Αφαιρούμε την κατάσταση 9 απο την ουρά

$V = \{1,2,3,4,5,8,9\}$

$Q = \{5,3\}$

Output = {1,2,4,8,9}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 5 και προσθέτουμε στην ουρά τις καταστάσεις - παιδιά της 5 , δηλαδή τις καταστάσεις 10 και 11.

$V = \{1,2,3,4,5,8,9,10,11\}$

$Q = \{10,11,3\}$

Output = {1,2,4,8,9,5}

Εφόσον το πρώτο στοιχείο της ουράς δηλαδή η κατάσταση 10 βρίσκεται στο επίπεδο 3, δεν εμβαθύνουμε άλλο. Αφαιρούμε την κατάσταση 10 απο την ουρά

$V = \{1,2,3,4,5,8,9,10,11\}$

$Q = \{11,3\}$

Output = {1,2,4,8,9,5,10}

Αφαιρούμε απο την ουρά το πρώτο στοιχείο δηλαδή την κατάσταση 11. Η κατάσταση 11 είναι ο στόχος της αναζήτησης, συνεπώς η αναζήτηση τερματίζεται.

$V = \{1,2,3,4,8,9,5,10,11\}$

$Q = \{3\}$

Output = {1,2,4,8,9,5,10,11}

Bidirectional Search

Με τη μέθοδο αυτή ξεκινάμε να διασχίζουμε το δέντρο απο την κατάσταση εκκίνησης και απο την κατάσταση-στόχο παράλληλα. Δηλαδή στον ίδιο γράφο διακρίνουμε 2 υποδέντρα τα οποία και διασχίζουμε παράλληλα. Η αναζήτηση σταματά όταν επεξεργαζόμαστε και στα 2 δέντρα κοινή κατάσταση/κόμβο. Προκειμένου να είναι εφικτή αυτού του είδους η αναζήτηση πρέπει:

α) Τόσο η κατάσταση εκκίνησης όσο και η κατάσταση-στόχος να είναι σαφώς ορισμένες (στο συγκεκριμένο πρόβλημα ισχύει)

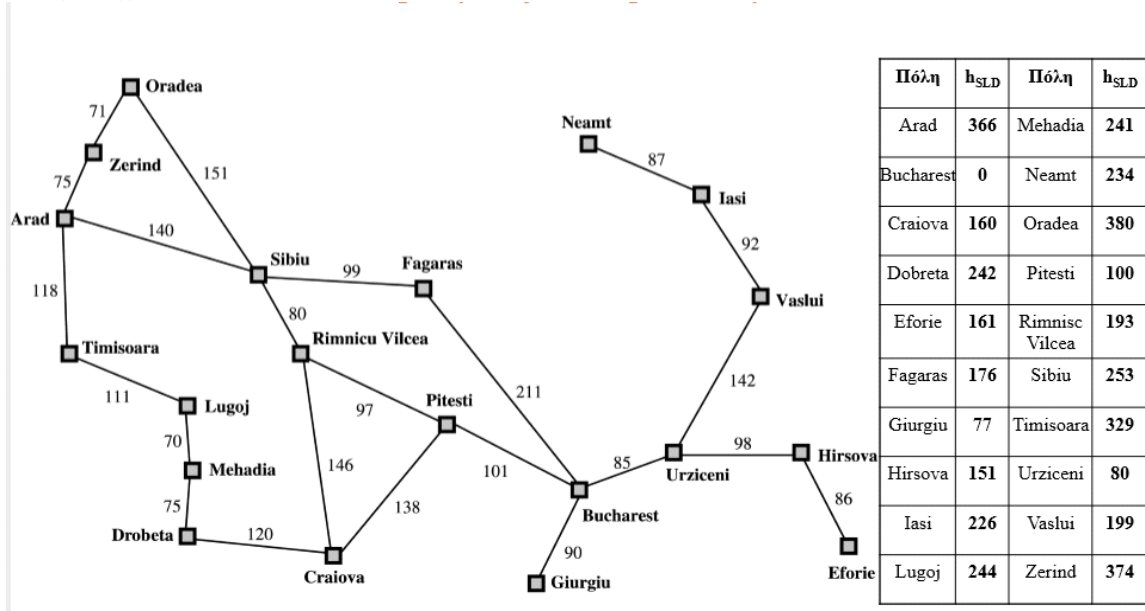
β) Τα δύο υποδέντρα να είναι της ίδιας μορφής (στο συγκεκριμένο πρόβλημα δεν ισχύει).

Επομένως δεν είναι εφικτή τέτοιου είδους αναζήτηση στο συγκεκριμένο γράφο.

Άσκηση 3η

Ζητούμενο είναι η εύρεση διαδρομής στον χάρτη της Ρουμανίας με αφετηρία την πόλη Lugoj και στόχο την πόλη Bucharest με χρήση του αλγόριθμου A*.

Ο αλγόριθμος A* αποτελεί έναν αλγόριθμο πληροφορημένης αναζήτησης. Σε κάθε βήμα υπολογίζουμε το άθροισμα $f(n) = g(n) + h(n)$, όπου g το κόστος διαδρομής απο μία πόλη σε μια άλλη και h η εκτιμώμενη απόσταση της εκάστοτε πόλης-προορισμού απο την πόλη-στοχο.



Ξεκινώντας απο την πόλη Lugoj πρέπει να αποφασίσουμε ποια διαδρομή θα ακολουθήσουμε. Υπολογίζουμε το κόστος των πιθανών διαδρομών απο την Lugoj προς τις πόλεις Timisoara και Mehadia.

$$F(\text{Timisoara}) = 111 + 329 = 440$$

$$F(\text{Mehadia}) = 70 + 241 = 311$$

Εφόσον $F(\text{Mehadia}) < F(\text{Timisoara})$ επιλέγουμε να επεκτείνουμε τις διαδρομές απο την πόλη Mehadia

Η μόνη διαθέσιμη διαδρομή απο την πόλη Mehadia είναι αυτήν προς την πόλη Drobeta με κόστος

$$F(\text{Drobeta}) = (70 \text{ (κόστος Lugoj προς Mehadia)} + 75) + 242 = 387$$

Επεκτείνοντας τις διαδρομές απο την πόλη Drobeta, η μόνη διαθέσιμη διαδρομή είναι αυτήν προς την πόλη Craiova με κόστος

$$F(\text{Craiova}) = (70(\text{κόστος Lugoj προς Mehadia}) + 75(\text{κόστος Mehadia προς Drobeta}) + 120) + 160 = 425$$

Από την πόλη Craiova υπάρχουν 2 διαθέσιμες διαδρομές, προς την πόλη Rimnicu Vilcea και προς την πόλη Pitesti.

$F(\text{Vilcea}) = (70(\text{κόστος Lugoι προς Mehadia}) + 75(\text{κόστος Mehadia προς Drobeta}) + 120(\text{κόστος Drobeta προς Craiova}) + 146) + 193 = 604$

$F(\text{Pitesti}) = (70(\text{κόστος Lugoι προς Mehadia}) + 75(\text{κόστος Mehadia προς Drobeta}) + 120(\text{κόστος Drobeta προς Craiova}) + 138) + 100 = 503$

Εφόσον $F(\text{Pitesti}) < F(\text{Vilcea})$ επιλέγουμε να επεκτείνουμε τις διαδρομές απο την πόλη Pitesti

Απο την πόλη Pitesti υπάρχουν 2 διαθέσιμες διαδρομές, προς την πολη Rimnicu Vilcea και προς την πόλη Bucharest.

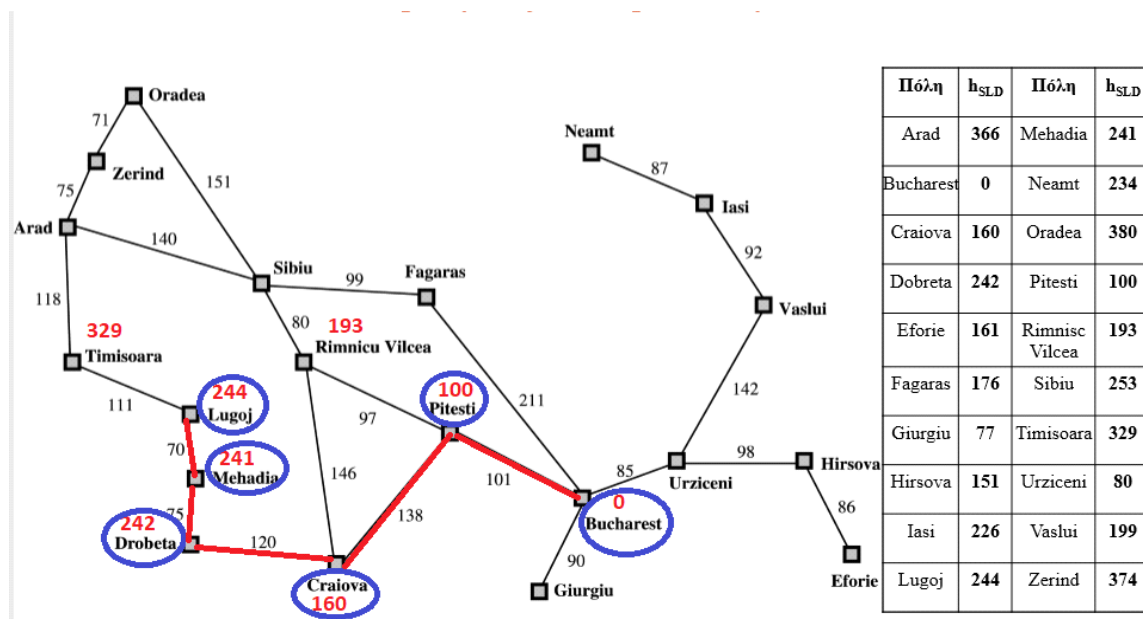
$F(\text{Vilcea}) = (70(\text{κόστος Lugoι προς Mehadia}) + 75(\text{κόστος Mehadia προς Drobeta}) + 120(\text{κόστος Drobeta προς Craiova}) + 138(\text{κόστος Craiova προς Pitesti}) + 97) + 193 = 693$

$F(\text{Bucharest}) = (70(\text{κόστος Lugoι προς Mehadia}) + 75(\text{κόστος Mehadia προς Drobeta}) + 120(\text{κόστος Drobeta προς Craiova}) + 138(\text{κόστος Craiova προς Pitesti}) + 101) + 0 = 504$

Εφόσον $F(\text{Bucharest}) < F(\text{Vilcea})$ επιλέγουμε να "επισκεφθούμε" την πόλη Bucharest.

Η πόλη Bucharest είναι η πόλη στόχος συνεπώς η αναζήτηση τερματίζεται, δίνοντας τελική διαδρομή :

Lugoι --> Mehadia --> Drobeta --> Craiova --> Pitesti --> Bucharest



Άσκηση 4η

α) Ακτινική Αναζήτηση για $k=1$:

Η ακτινική αναζήτηση πραγματοποιεί παράλληλη αναρρίχηση λόφων από k διαφορετικές αρχικές καταστάσεις. Για $k=1$ ο αλγόριθμος ταυτίζεται με τον αλγόριθμο αναρρίχησης λόφων.

β) Προσομοιωμένη ανόπτηση με $T=0$:

Η προσομοιωμένη ανόπτηση αποτελεί μια μορφή στοχαστικής αναρρίχησης λόφων. Για $T=0$ ο αλγόριθμος παραμένει σταθερά στην αρχική κατάσταση.

γ) Γενετικός Αλγόριθμος με μέγεθος πληθυσμού $N=1$:

Ο γενετικός αλγόριθμος αποτελεί επίσης μια μορφή στοχαστικής αναρρίχησης λόφων. Για $N=1$ ο αλγόριθμος ταυτίζεται με τον αλγόριθμο αναρρίχησης λόφων.