ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΜΑΘΗΜΑ: ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ Ι

Εργασία 1

Δημοσθένης Θεοδοσίου (1115202300051)

Οκτώβριος 2024

Πρόβλημα 2:

Έχετε το εξής πρόβλημα αναζήτησης:

- Ο χώρος καταστάσεων αναπαρίσταται ως δένδρο.
- Ο κόμβος της ρίζας (αρχική κατάσταση) έχει τρεις κόμβους-παιδιά.
- Κάθε ένας από αυτούς τους κόμβους-παιδιά έχει επίσης τρεις κόμβους-παιδιά κ.ο.κ. Δηλαδή, το δένδρο έχει ομοιόμορφο παράγοντα διακλάδωσης ίσο με 3.
- Ο στόχος βρίσκεται στο βάθος 4.

Να υπολογίσετε θεωρητικά τον μικρότερο και το μεγαλύτερο αριθμό κόμβων που επεκτείνονται από κάθε έναν από τους παρακάτω αλγόριθμους αναζήτησης, υποθέτοντας ότι εκτελούν πλήρη αναζήτηση (δηλαδή, μέχρι να βρεθεί ο στόχος):

- Αναζήτηση πρώτα κατά πλάτος (BFS)
- Αναζήτηση πρώτα κατά βάθος (DFS). Υποθέστε ότι ο DFS εξετάζει πάντα πρώτα το αριστερότερο παιδί.
- Αναζήτηση με επαναληπτική εκβάθυνση (IDS).

Απάντηση:

Στο πρόβλημα δεν διευκρινίζεται εάν ο χώρος καταστάσεων είναι πεπερασμένος ή άπειρος. Στην λύση μου θα εξετάσω όλες τις δυνατές περιπτώσεις. Επίσης θεωρώ πως η αρίθμιση των επιπέδων ξεκινάει από το 0. Από τα δεδομένα του προβλήματος προκύπτει πως b = 3 και d = 4.

- Αναζήτηση πρώτα κατά πλάτος (BFS)

Γενικά, ο BFS σε γράφους έχει πολυπλοκότητα $O(b^{(d+1)})$, αλλά θα θεωρήσουμε πως έχουμε την βελτιωμένη έκδοση με μικρότερη πολυπλοκότητα $O(b^d)$. Η καλύτερη περίπτωση είναι ο κόμβος στόχος να βρίσκεται στο αριστερότερο σημείο του επιπέδου με βάθος 4. Αυτό προκύπτει αν αναπτύξουμε όλα τα επίπεδα μέχρι το 3 και αναπτύξουμε τον πρώτο κόμβο του τέταρτου επιπέδου. Επειδή κάθε επίπεδο έχει $O(b^{(\alpha/\alpha\})})$, έχουμε συνολικά 1+3+9+27+1=41 κόμβους. Τώρα η χειρότερη περίπτωση είναι ο κόμβος στόχος να βρίσκεται στο δεξιότερο σημείο του τέταρτου επιπέδου. Έτσι προκύπτουν 1+3+9+27+81=121 κόμβοι. Βλέπουμε πως στο τρέχον πρόβλημα δεν έχει επιρροή στον BFS η περατότητα ή μη του χώρου καταστάσεων.

- Αναζήτηση πρώτα κατά βάθος (DFS)

  Γενικά ο αλγόριθμος DFS έχει πολυπλοκότητα $O(b^{\mu})$.

  - Περίπτωση 1 : Μη πεπερασμένος χώρος καταστάσεων

    Στην περίπτωση που ο χώρος καταστάσεων δεν είναι πεπερασμένος και ο DFS εξετάζει πάντα το αριστερότερο παιδί, έχουμε δύο ενδεχόμενα. Εάν ο κόμβος στόχος είναι το αριστερότερο παιδί του 4ου επιπέδου, ο DFS θα χρειαστεί να αναπτύξει μόλις 5 κόμβους. Σε κάθε άλλη περίπτωση όπου ο κόμβος στόχος δεν είναι το αριστερότερο στοιχείο του επιπέδου, ο αλγόριθμος θα αναπτύξει άπειρους κόμβους.

  - Περίπτωση 2 : Πεπερασμένος χώρος καταστάσεων

    Στην περίπτωση που ο χώρος καταστάσεων είναι πεπερασμένος και ο DFS εξετάζει πάντα το αριστερότερο παιδί, έχουμε δύο ενδεχόμενα. Εάν ο κόμβος στόχος είναι το αριστερότερο παιδί του

4ου επιπέδου, ο DFS θα χρειαστεί να αναπτύξει μόλις 5 κόμβους. Αυτή είναι η καλύτερη περίπτωση. Η χειρότερη περίπτωση είναι όταν ο κόμβος βρίσκεται πάλι στο δεξιότερο σημείο του τέταρτου επιπέδου. Εκεί δεν θα αναπτυχθούν μόνο $1 + 3 + 9 + 27 = 40$ κόμβοι, όπως θα γινόταν εάν όλοι οι κόμβοι πήγαιναν μέχρι και το 4ο επίπεδο, αλλά στην πραγματικότητα θα αναπτυχθούν $40 + ξ$ κόμβοι. Το ξ εδώ αναπαριστά το άθροισμα των μεγεθών όλων εκείνων των υποδέντρων τα οποία έχουν ρίζα όλους τους κόμβους του 4ου επιπέδου πλην του τελευταίου.

- Αναζήτηση με επαναληπτική εκβάθυνση (IDS)

  Εφόσον ο IDS είναι μία σύζευξη του DFS και του BFS, είναι ασφαλές να υποθέσουμε ότι για κάθε επίπεδο που εξετάζει ο IDS το κάνει εξετάζοντας πάντοτε το αριστερότερο παιδί πρώτα. Πάλι εδώ, η καλύτερη περίπτωση είναι το αριστερότερο παιδί του 4ου επιπέδου. Εδώ θα χρειαστεί ο IDS να αλλάξει βάθος 5 φορές, δηλαδή θα τρέξει DLS 5 φορές, κάθε μία με διαφορετικό όριο βάθους. Για το πρώτο DLS 1 κόμβο, για το 2ο (1+3) κόμβους, για το 3ο DLS (1+3+9) κόμβους, για το 4ο DLS (1+3+9+27) κόμβους, για το 5ο 5 κόμβους. Άρα στην καλύτερη περίπτωση θα αναπτύξει 63 κόμβους. Στην χειρότερη περίπτωση, ο κόμβος στόχος θα βρίσκεται στο δεξιότερο σημείο του 4ου επιπέδου. Άρα στο τελευταίο DLS του θα αναπτύξει (1+3+9+27+81) κόμβους. Αυτό μαζί με τα προηγούμενα DLS μας δίνει 179 αναπτυγμένους κόμβους. Βλέπουμε εδώ πάλι ότι δεν κάνει διαφορά η περαρότητα ή μη του χώρου, αφού ο IDS έχει χαρακτηριστικά του BFS, χάρη στα οποία αποφεύγει να αναπτύξει άπειρους κόμβους.

Πρόβλημα 3:

Θεωρήστε το παρακάτω πρόβλημα αναζήτησης. Ένα ρομπότ έχει αναλάβει να παραδώσει ένα δέμα από μια αποθήκη σε ένα προκαθορισμένο σημείο παράδοσης σε μια πόλη της οποίας ο χάρτης αναπαρίσταται ως 2Δ πλέγμα. Το ρομπότ μπορεί να κινηθεί προς τέσσερις κατευθύνσεις: πάνω, κάτω, αριστερά και δεξιά. Κάθε κίνηση έχει ένα κόστος που εξαρτάται από τον τύπο του εδάφους ή του δρόμου που πρέπει να διασχίσει το ρομπότ. Στόχος σας είναι να βοηθήσετε το ρομπότ να βρει τη βέλτιστη διαδρομή από την αποθήκη στο σημείο παράδοσης χρησιμοποιώντας τον αλγόριθμο A*.

Το πλέγμα της πόλης περιέχει διάφορους τύπους εδάφους:

- S: Αφετηρία (Αποθήκη)
- G: Στόχος (Σημείο παράδοσης)
- R: Κανονικός δρόμος (κόστος 1)
- H: Αυτοκινητόδρομος (κόστος 0.5)
- B: Κτίριο (αδιάβατο εμπόδιο)
- P: Πάρκο (κόστος 2)
- W: Νερό (αδιάβατο εμπόδιο)

Το 10x10 πλέγμα είναι το εξής:

$$\begin{array}{cccccccccc}
S & R & R & R & B & W & R & H & H & H \\
R & B & B & R & H & H & R & R & B & H \\
R & P & P & R & B & R & R & R & B & R \\
R & R & R & R & W & R & P & P & R & R \\
R & R & B & R & R & R & H & H & R & B \\
B & W & R & P & P & R & B & R & R & R \\
P & P & R & R & R & R & R & R & B & B \\
R & B & R & R & R & W & H & H & R & R \\
R & R & R & R & B & R & R & R & B & R \\
H & H & H & B & B & R & R & G & R & R
\end{array}$$

Το ρομπότ πρέπει να βρει τη διαδρομή από την αποθήκη (Σ) στο σημείο παράδοσης (Γ) που ελαχιστοποιεί το συνολικό κόστος μετακίνησης.

Να υπολογίσετε το συνολικό κόστος της βέλτιστης διαδρομής (άθροισμα του κόστους των κελιών που διασχίζονται) που θα βρεθεί από τον A* αν τον χρησιμοποιήσουμε στο πρόβλημα με ευρετική συνάρτηση την απόσταση Manhattan διά 2. Δηλαδή το αποτέλεσμα της ευρετικής για τη μετάβαση από έναν κόμβο x σε έναν κόμβο y, δίνεται από τον παρακάτω τύπο:

$$h(x, y) = \frac{\text{ManhattanDistance}(x, y)}{2}$$

Επίσης να υπολογίσετε:

- Τον αριθμό των κόμβων που επεκτάθηκαν από τον A* κατά τη διάρκεια της αναζήτησης.
- Τη σειρά με την οποία βγαίνουν οι κόμβοι από τη λίστα «σύνορο» (fringe).

Να δώσετε την ενέργεια του ρομπότ που αντιστοιχεί σε κάθε κόμβο.
Να υποθέσετε ότι:

- (α) οι διαθέσιμες ενέργειες του ρομπότ εφαρμόζονται με τη σειρά που δόθηκαν παραπάνω, και
- (β) όταν ο A* δεν μπορεί να «διακρίνει» δύο κόμβους, τότε επιλέγει τον αριστερότερο στο δένδρο αναζήτησης.

Τέλος, να προτείνετε δύο επιπλέον παραδεκτές ευρετικές συναρτήσεις για τη λύση του παραπάνω προβλήματος χρησιμοποιώντας τον αλγόριθμο A*. (Δεν χρειάζεται να λύσετε το πρόβλημα χρησιμοποιώντας τις, μόνο να τις ορίσετε και να αποδείξετε ότι είναι παραδεκτές.)

Απάντηση:

Για την αναπαράσταση της λύσης θα κάνουμε τις εξής παραδοχές: κάθε κόμβος αναπαρίσταται σαν tuple (x , y) , για κάθε άξονα. Στο πρόβλημα μας x είναι ο κάθετος άξονας και y ο παράλληλος στο επίπεδο άξονας. Η αρίθμιση ξεκιννάει από το 0. Για την λύση μας θα έχουμε δύο δομές ένα PriorityQueue και μια list. Το PriorityQueue θα λέγεται fringe και θα περιλαμβάνει μόνο tuple μορφής (x,y real_cost, heuristic_cost , move). Η λίστα θα περιλαμβάνει μια συλλογή από tuples, όπου κάθε tuple περιέχει τις συντεταγμένες x , y , το πραγματικό ελάχιστο κόστος για να φτάσουμε σε αυτόν τον κόμβο, καθώς και την κίνηση που χρειάστηκε από τον πατέρα για να φτάσουμε σε αυτόν τον κόμβο. Συνεπώς θα είναι της μορφής (x , y , real cost , move). Επίσης υποθέτω ότι το κόστος για να πάω στον κόμβο στόχο είναι 1.

- 1ο βήμα
  Fringe: [(0 , 0 , 0 , 8, 0)]
  Explored: [ ]
- 2ο βήμα
  Fringe: [(1,0,1,8.5,K), (0,1,1,8.5,D)]
  Explored: [(0,0,0,0)]
- 3ο βήμα
  Fringe: [(0,1,1,8.5,D) (2,0,2,9,K)]
  Explored: [(0,0,0,0) , (1,0,1,K)]
- 4ο βήμα
  Fringe: [(2,0,2,9,K) , (0,2,2,9,R)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D)]
- 5ο βήμα
  Fringe: [(0,2,2,9,D) , (3,0,3,9.5,K) , (2,1,4,10.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K)]
- 6ο βήμα
  Fringe: [(3,0,3,9.5,K) , (0,3,3,9.5,K) , (2,1,4,10.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D)]
- 7ο βήμα
  Fringe: [(0,3,3,9.5,K) ,(4,0,4,10,K) , (3,1,4,10,D) (2,1,4,10.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K)]
- 8ο βήμα
  Fringe: [(4,0,4,10,K) , (3,1,4,10,D), (1,3,4,10,K),(2,1,4,10.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K)]
- 9ο βήμα
  Fringe: [(3,1,4,10,D), (1,3,4,10,K),(2,1,4,10.5,D) , (4,1,5,10.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K)]
- 10ο βήμα
  Fringe: [(1,3,4,10,K),(2,1,4,10.5,D) , (4,1,5,10.5,D) ,(3,2,5,10.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D)]

- 11ο βήμα
  Fringe: [(1,4,4.5,10,K),(2,1,4,10.5,D) , (4,1,5,10.5,D) ,(3,2,5,10.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K)]

- 12ο βήμα
  Fringe: [(1,5,5,10,K),(2,1,4,10.5,D) , (4,1,5,10.5,D) ,(3,2,5,10.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K)]

- 13ο βήμα
  Fringe: [(2,1,4,10.5,D) , (4,1,5,10.5,D) ,(3,2,5,10.5,D), (2,5,6,10.5,K) , (1,6,6,10.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K)]

- 14ο βήμα
  Fringe: [(4,1,5,10.5,D) ,(3,2,5,10.5,D), (2,5,6,10.5,K) , (1,6,6,10.5,D), (2,2,6,12,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D)]

- 15ο βήμα
  Fringe: [(3,2,5,10.5,D), (2,3,5,10.5 ,K), (2,5,6,10.5,K) , (1,6,6,10.5,D), (2,2,6,12,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D)]

- 16ο βήμα
  Fringe: [(2,3,5,10.5 ,K), (2,5,6,10.5,K) , (1,6,6,10.5,D), (3,3,6,11,D) (2,2,6,12,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D)]

- 17ο βήμα
  Fringe: [(2,5,6,10.5,K) , (1,6,6,10.5,D), (3,3,6,11,D) (2,2,6,12,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K)]

- 18ο βήμα
  Fringe: [(1,6,6,10.5,D), (3,3,6,11,D) , (3,5,7,11,K) , (2,6,7,11,D) (2,2,6,12,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K)]

- 19ο βήμα
  Fringe: [(3,3,6,11,D) , (3,5,7,11,K) , (2,6,7,11,D) , (1,7,7,11,D) (2,2,6,12,D) , (0,6,7,12,P)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D)]

- 20ο βήμα
  Fringe: [(3,5,7,11,K) , (2,6,7,11,D) , (1,7,7,11,D),(4,3,7,11.5,K) (2,2,6,12,D) , (0,6,7,12,P)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D)]

- 21ο βήμα
  Fringe: [(2,6,7,11,D) , (1,7,7,11,D),(4,3,7,11.5,K),(4,5,8,11.5,K) (2,2,6,12,D) , (0,6,7,12,P) ,(3,6,9,12.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K)]

- 22ο βήμα
  Fringe: [(1,7,7,11,D),(4,3,7,11.5,K),(4,5,8,11.5,K),(2,7,8,11.5,D) (2,2,6,12,D) , (0,6,7,12,P) ,(3,6,9,12.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D)]

- 23ο βήμα
  Fringe: [(4,3,7,11.5,K),(4,5,8,11.5,K),(2,7,8,11.5,D) (2,2,6,12,D) , (0,6,7,12,P),(0,7,7.5,12,P) ,(3,6,9,12.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D)]

- 24ο βήμα
Fringe: [(4,5,8,11.5,K),(2,7,8,11.5,D) (2,2,6,12,D) , (0,6,7,12,P),(0,7,7.5,12,P),(4,4,8,12,D),(3,6,9,12.5,D) ,(5,3,9,13,K)]
Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K)]

- 25ο βήμα
Fringe: [(2,7,8,11.5,D),(4,6,8.5,11.5.D) (2,2,6,12,D) , (0,6,7,12,P),(0,7,7.5,12,P),(4,4,8,12,D),(5,5,9,12,K) ,(3,6,9,12.5,D),(5,3,9,13,K)]
Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K)]

- 26ο βήμα
Fringe: [(4,6,8.5,11.5.D) (2,2,6,12,D) , (0,6,7,12,P),(0,7,7.5,12,P),(4,4,8,12,D),(5,5,9,12,K),(3,6,9,12.5,D) ,(5,3,9,13,K),(3,7,10,13,K)]
Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D)]

- 27ο βήμα
Fringe: [(4,7,9,11.5,D) (2,2,6,12,D) , (0,6,7,12,P),(0,7,7.5,12,P),(4,4,8,12,D),(5,5,9,12,K),(3,6,9,12.5,D) ,(5,3,9,13,K),(3,7,10,13,K)]
Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D)]

- 28ο βήμα
Fringe: [(2,2,6,12,D) , (0,6,7,12,P),(0,7,7.5,12,P),(4,4,8,12,D),(5,5,9,12,K),(5,7,10,12,K),(3,6,9,12.5,D) ,(5,3,9,13,K),(3,7,10,13,K),(4,8,10,13,D)]
Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D)]

- 29ο βήμα
Fringe: [(0,6,7,12,P),(0,7,7.5,12,P),(4,4,8,12,D),(5,5,9,12,K),(5,7,10,12,K)(3,6,9,12.5,D),(5,3,9,13,K) ,(3,7,10,13,K),(4,8,10,13,D)]
Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D)]

- 30ο βήμα
Fringe: [(0,7,7.5,12,P),(4,4,8,12,D),(5,5,9,12,K),(5,7,10,12,K)(3,6,9,12.5,D),(5,3,9,13,K),(3,7,10,13,K) ,(4,8,10,13,D)]
Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P)]

- 31ο βήμα
Fringe: [(4,4,8,12,D),(5,5,9,12,K),(5,7,10,12,K)(3,6,9,12.5,D),(5,3,9,13,K),(3,7,10,13,K),(4,8,10,13,D) ,(0,8,10,13,D)]
Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P)]

- 32ο βήμα
Fringe: [(5,5,9,12,K),(5,7,10,12,K)(3,6,9,12.5,D),(5,3,9,13,K),(3,7,10,13,K),(4,8,10,13,D),(0,8,10,13,D) ,(5,4,10.5,13.5,K)]
Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D)]

- 33ο βήμα
  Fringe: [(5,7,10,12,K),(3,6,9,12.5,D),(6,5,10,12.5,K),(5,3,9,13,K),(3,7,10,13,K),(4,8,10,13,D),(0,8,10,13,D) ,(5,4,10.5,13.5,K)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K)]

- 34ο βήμα
  Fringe: [(3,6,9,12.5,D),(6,5,10,12.5,K),(6,7,11,12.5,K),(5,3,9,13,K),(3,7,10,13,K),(4,8,10,13,D),(0,8,10,13,D) ,(5,4,10.5,13.5,K),(5,8,11,13.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K)]

- 35ο βήμα
  Fringe: [(6,5,10,12.5,K),(6,7,11,12.5,K),(5,3,9,13,K),(3,7,10,13,K),(4,8,10,13,D),(0,8,10,13,D),(5,4,10.5,13.5,K) ,(5,8,11,13.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D)]

- 36ο βήμα
  Fringe: [(6,7,11,12.5,K),(5,3,9,13,K),(3,7,10,13,K),(4,8,10,13,D),(0,8,10,13,D),(6,6,11,13,D),(5,4,10.5,13.5,K) ,(5,8,11,13.5,D),(6,4,11,14,A)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K)]

- 37ο βήμα
  Fringe: [(7,7,11.5,12.5,K)(5,3,9,13,K),(3,7,10,13,K),(4,8,10,13,D),(0,8,10,13,D),(6,6,11,13,D),(5,4,10.5,13.5,K) ,(5,8,11,13.5,D),(6,4,11,14,A)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K),(6,7,11,K)]

- 38ο βήμα
  Fringe: [(5,3,9,13,K),(3,7,10,13,K),(4,8,10,13,D),(0,8,10,13,D),(6,6,11,13,D),(8,7,12.5,13,K),(5,4,10.5,13.5,K) ,(5,8,11,13.5,D),(7,6,12,13.5,A),(6,4,11,14,A),(7,8,12.5,14,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K),(6,7,11,K),(7,7,11.5,K)]

- 39ο βήμα
  Fringe: [(3,7,10,13,K),(4,8,10,13,D),(0,8,10,13,D),(6,6,11,13,D),(8,7,12.5,13,K),(5,4,10.5,13.5,K),(5,8,11,13.5,D) ,(7,6,12,13.5,A),(6,3,10.5,13.5,K),(6,4,11,14,A),(7,8,12.5,14,D),(5,2,10,14.5,A)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K),(6,7,11,K),(7,7,11.5,K), (5,3,9,K)]

- 40ο βήμα
  Fringe: [(4,8,10,13,D),(0,8,10,13,D),(6,6,11,13,D),(8,7,12.5,13,K),(5,4,10.5,13.5,K),(5,8,11,13.5,D),(7,6,12,13.5,A) ,(6,3,10.5,13.5,K),(6,4,11,14,A),(7,8,12.5,14,D),(5,2,10,14.5,A),(3,8,11,14.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K),(6,7,11,K),(7,7,11.5,K), (5,3,9,K), (3,7,10,K)]

- 41ο βήμα
  Fringe: [(0,8,10,13,D),(6,6,11,13,D),(8,7,12.5,13,K),(5,4,10.5,13.5,K),(5,8,11,13.5,D),(7,6,12,13.5,A)

,(6,3,10.5,13.5,K),(6,4,11,14,A),(7,8,12.5,14,D),(5,2,10,14.5,A),(3,8,11,14.5,D)]
Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K),(6,7,11,K),(7,7,11.5,K), (5,3,9,K), (3,7,10,K),(4,8,10,D)]

- 42ο βήμα
  Fringe: [(6,6,11,13,D),(8,7,12.5,13,K),(5,4,10.5,13.5,K),(5,8,11,13.5,D),(7,6,12,13.5,A),(6,3,10.5,13.5,K) ,(6,4,11,14,A),(7,8,12.5,14,D),(0,9,8.5,14,D),(5,2,10,14.5,A),(3,8,11,14.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K),(6,7,11,K),(7,7,11.5,K), (5,3,9,K), (3,7,10,K),(4,8,10,D), (0,8,10,D)]

- 43ο βήμα
  Fringe: [(8,7,12.5,13,K),(7,6,11.5,13,K),(5,4,10.5,13.5,K),(5,8,11,13.5,D),(6,3,10.5,13.5,K),(6,4,11,14,A) ,(7,8,12.5,14,D),(0,9,8.5,14,D),(5,2,10,14.5,A),(3,8,11,14.5,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K),(6,7,11,K),(7,7,11.5,K), (5,3,9,K), (3,7,10,K),(4,8,10,D), (0,8,10,D),(6,6,11,D)]

- 44ο βήμα
  Fringe: [(7,6,11.5,13,K),(5,4,10.5,13.5,K),(5,8,11,13.5,D),(6,3,10.5,13.5,K),(9,7,13.5,13.5,K),(6,4,11,14,A) ,(7,8,12.5,14,D),(0,9,8.5,14,D),(5,2,10,14.5,A),(3,8,11,14.5,D),(8,6,13.5,14.5,A)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K),(6,7,11,K),(7,7,11.5,K), (5,3,9,K), (3,7,10,K),(4,8,10,D), (0,8,10,D),(6,6,11,D), (8,7,12.5,K)]

- 45ο βήμα
  Fringe: [(5,4,10.5,13.5,K),(5,8,11,13.5,D),(6,3,10.5,13.5,K),(9,7,13.5,13.5,K),(8,6,12.5,13.5,A),(6,4,11,14,A) ,(7,8,12.5,14,D),(0,9,8.5,14,D),(5,2,10,14.5,A),(3,8,11,14.5,D),]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K),(6,7,11,K),(7,7,11.5,K), (5,3,9,K), (3,7,10,K),(4,8,10,D), (0,8,10,D),(6,6,11,D), (8,7,12.5,K),(7,6,11.5,K)]

- 46ο βήμα
  Fringe: [(5,8,11,13.5,D),(6,3,10.5,13.5,K),(9,7,13.5,13.5,K),(8,6,12.5,13.5,A),(6,4,11,14,A),(7,8,12.5,14,D) ,(0,9,8.5,14,D),(5,2,10,14.5,A),(3,8,11,14.5,D),]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K),(6,7,11,K),(7,7,11.5,K), (5,3,9,K), (3,7,10,K),(4,8,10,D), (0,8,10,D),(6,6,11,D), (8,7,12.5,K),(7,6,11.5,K),(5,4,10.5,K)]

- 47ο βήμα
  Fringe: [(6,3,10.5,13.5,K),(9,7,13.5,13.5,K),(8,6,12.5,13.5,A),(6,4,11,14,A),(7,8,12.5,14,D),(0,9,8.5,14,D) ,(5,2,10,14.5,A),(3,8,11,14.5,D),(5,9,12,15,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K),(6,7,11,K),(7,7,11.5,K), (5,3,9,K), (3,7,10,K),(4,8,10,D), (0,8,10,D),(6,6,11,D), (8,7,12.5,K),(7,6,11.5,K),(5,4,10.5,K),(5,8,11,D)]

- 48ο βήμα
  Fringe: [(9,7,13.5,13.5,K),(8,6,12.5,13.5,A),(6,4,11,14,A),(7,8,12.5,14,D),(0,9,8.5,14,D),(7,3,11,14,K) ,(5,2,10,14.5,A),(3,8,11,14.5,D),(5,9,12,15,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K),(6,7,11,K),(7,7,11.5,K), (5,3,9,K), (3,7,10,K),(4,8,10,D), (0,8,10,D),(6,6,11,D), (8,7,12.5,K),(7,6,11.5,K),(5,4,10.5,K),(5,8,11,D), (6,3,10.5,K)]

Στο τελευταίο βήμα απλά βγάζουμε από το φρινγε τον πρώτο κόμβο και βρήκαμε κατάσταση Γοαλ, οπότε και τερματίζουμε τον αλγόριθμο.

- 49ο βήμα
  Fringe: [(8,6,12.5,13.5,A),(6,4,11,14,A),(7,8,12.5,14,D),(0,9,8.5,14,D),(7,3,11,14,K),(5,2,10,14.5,A) ,(3,8,11,14.5,D),(5,9,12,15,D)]
  Explored: [(0,0,0,0) , (1,0,1,K) , (0,1,1,D) , (2,0,2,K) , (0,2,2,D) , (3,0,3,K) , (0,3,3,K) , (4,0,4,K) , (3,1,4,D) , (1,3,4,K), (1,4,4.5,K) , (1,5,5,K) , (2,1,4,D) , (4,1,5,D) , (3,2,5,D) ,(2,3,5,K) , (2,5,6,K) , (1,6,6,D) , (3,3,6,D) , (3,5,7,K) , (2,6,7,D) , (1,7,7,D),(4,3,7,K),(4,5,8,K),(2,7,8,D),(4,6,8.5,D),(4,7,9,D), (2,2,6,D),(0,6,7,P),(0,7,7.5,P),(4,4,8,D),(5,5,9,K),(5,7,10,K),(3,6,9,D),(6,5,10,K),(6,7,11,K),(7,7,11.5,K), (5,3,9,K), (3,7,10,K),(4,8,10,D), (0,8,10,D),(6,6,11,D), (8,7,12.5,K),(7,6,11.5,K),(5,4,10.5,K),(5,8,11,D), (6,3,10.5,K),(9,7,13.5,K)]

Επομένως χρειάστηκε να κάνουμε expand 47 κόμβους, μέχρις ότου το goal state να έρθει ως πρώτο στοιχείο του fringe. Επομένως το ελάχιστο κόστος για τον στόχο είναι 13.5 και επιτυγχάνεται με την διαδρομή : [Δ, Δ, Δ, Κ, Δ, Δ, Κ, Κ, Κ, Δ, Δ, Κ, Κ, Κ, Κ, Κ].

Σε ότι αφορά το κομμάτι των ευρετικών, έχουμε 2 εναλλακτικές. Η πρώτη είναι απλά η ευκλείδια ευρετική που υπολογίζει την διαγώνια απόσταση ανάμεσα στο σημείο έναρξης και το σημείο εκκίνησης. Η συγκεκριμένη ευρετική είναι σίγουρα admissible, αφού πάντα σε οποιοδήποτε ορθογώνιο τρίγωνο η υποτείνουσα είναι μικρότερη από το άθροισμα των προσκείμενων πλευρών του τριγώνου. Η δεύτερη ευρετική, αν και λιγότερο ρεαλιστική είναι να παίρνουμε την Manhattan Distance για μία μόνο διάσταση αντί για τις δύο. Δηλαδή μόνο για τον άξονα x/y.

Πρόβλημα 4

Θεωρήστε τον αλγόριθμο αμφίδρομης αναζήτησης που παρουσιάσαμε στις διαλέξεις. Θεωρείστε ότι στα προβλήματα αναζήτησης που θα εφαρμοστεί ο αλγόριθμος υπάρχει μοναδική κατάσταση στόχου. Υποθέτουμε ότι τα ζευγάρια αλγορίθμων που χρησιμοποιεί η αμφίδρομη αναζήτηση σαν υπορουτίνες για την (προς τα εμπρός) αναζήτηση από την αρχική κατάσταση και την (προς τα πίσω) αναζήτηση από την κατάσταση στόχου είναι:

(α) Αναζήτηση πρώτα σε πλάτος και αναζήτηση περιορισμένου βάθους

(β) Αναζήτηση με επαναληπτική εκβάθυνση και αναζήτηση περιορισμένου βάθους

(γ) A* και αναζήτηση περιορισμένου βάθους

(δ) A* και A*

Είναι ο αλγόριθμος αμφίδρομης αναζήτησης με υπορουτίνες όπως στα (α)-(δ) πλήρης· Είναι βέλτιστος· Ναι ή όχι και υπό ποιες συνθήκες. Πως μπορεί να γίνει αποδοτικά ο έλεγχος ότι οι δύο αναζητήσεις συναντιούνται σε κάθε μια από τις παραπάνω περιπτώσεις (α)-(δ)·

Απάντηση:

- Αναζήτηση πρώτα σε πλάτος και αναζήτηση περιορισμένου βάθους:

  Με βάση την συνάρτηση που μας δόθηκε στο φροντιστήριο ο αλγόριθμος θα αναπτύσσει συνέχεια κόμβους μέσω του DLS. Αυτό συμβαίνει γιατί ο DLS παράγει αρνητικές τιμές για το frontier γιατί είναι τύπου DFS. Εάν για τον DLS επιλέξουμε πολύ κοντό βάθος τότε θα αναπτύσσει κόμβους μέχρι να φτάσει τα όρια του. Έπειτα θα αρχίσει να αναπτύσει κόμβους ο BFS και συνεπώς θα βρεί μετά την λύση γιατί είναι πλήρης. Τώρα εάν για τον DLS επιλέξουμε βάθος μεγαλύτερο από το βάθος της λύσης, τότε απλά ο DLS θα αναπτύσσει συνέχεια κόμβους μέχρι να βρεί τον κόμβο στόχο. Επομένως αυτός ο συνδυασμός αλγορίθμων είναι πλήρης στην περίπτωση που ο παράγοντας διακλάδωσης είναι πεπερασμένος. Εάν δεν είναι τότε δεν θα βρεθεί ποτέ λύση. Προφανώς και αυτός ο συνδυαασμός δεν παράγει βέλτιστη λύση γιατί δεν λαμβάνει υπόψη παραμέτρους όπως τα διαφορετικά κόστη κάποιον μονοπατιών.

- Αναζήτηση με επαναληπτική εκβάθυνση και αναζήτηση περιορισμένου βάθους:

  Με βάση πάλι την συνάρτηση του φροντιστηρίου και οι δύο αλγόριθμοι θα παράγουν αρνητικές τιμές για το frontier, όμως επειδή στην αρχή ξεκινούν με 0 θα καλείται συνέχεια ο DLS. Εάν επιλέξουμε πάλι κοντό βάθος θα εξερευνήσει μέχρι ένα σημείο και μετά θα ξεκινήσει να ψάχνει ο IDS και αυξάνοντας σιγά σιγά το βάθος θα καταφέρει να βρεί την λύση. Τώρα βάλουμε βάθος στον DLS μεγαλύτερο από τον βάθος της διαδρομής στόχου, τότε απλά θα αναπτύσσει πάλι κόμβους μέχρι να βρεί τον στόχο. Άρα πάλι σε γενικές γραμμές ο συνδυασμός είναι πλήρης. Εάν όμως ο παράγοντας διακλάδωσης δεν είναι πεπερασμένος δεν θα μπορέσουμε να βρούμε λύση. Ο συνδυασμός αυτός πάλι δεν μας εγγυάται βέλτιστη λύση, αφού δεν τον ενδιαφέρουν τα κόστη των διαδρομών, αλλά οι ρηχές διαδρομές.

- A* και αναζήτηση περιορισμένου βάθους:

  Πάλι σε αυτή την περίπτωση για τους ίδιους λόγους με παραπάνω θα αρχίσει να προχωράει ο DLS πρώτος. Εάν πάρουμε κοντό βάθος θα πάει μέχρι ένα σημείο και μετά την λύση θα την βρεί ο A*. Ακόμη βάθος μεγαλύτερο τους βάθους λύσης να βάλουμε ο DLS θα καταφέρει να βρεί λύση. Πάλι σε γενικές γραμμές θα είναι πλήρης ο συνδυασμός, αρκεί ο παράγοντας διακλάδωσης να είναι πεπερασμένος. Η λύση που θα παράγεται δεν είναι βέλτιστη. Μέχρι το σημείο που συναντιούνται οι αλγόριθμοι ο A* ακολουθεί καλό μονοπάτι, αλλά από το σημείο και μετά που ακολουθούμε την διαδρομή που χάραξε ο DLS, δεν μας εγγυάται κανείς ότι πάμε με το βέλτιστο μονοπάτι.

- A* και A*:

  Ο τελευταίος συνδυασμός είναι πλήρης προφανώς γιατί είναι και οι δύο A*. Στην περίπτωση μη πεπερασμένου παράγοντα διακλάδωσης όμως παύει να είναι πλήρης. Πάντως σε γενικές γραμμές με admissible ευρετικές πάντα συναντιώονται και μάλιστα παράγουν την βέλτιστη λύση, γιατί λαμβάνουν υπόψη τα διαφορετικά κόστη κάθε κόμβου και κινούνται κάθε φορά προς την διαδρομή με το μικρότερο κόστος.

Ο καλύτερος τρόπος για να ελέγχουμε αν οι δύο αναζητήσεις συναντιούνται αποδοτικά σε όλες τις παραπάνω περιπτώσεις, είναι για κάθε αλγόριθμο να δημιουργούμε ένα set που υλοποιείται με hashmaps και αποθηκεύει όλους τους κόμβους που έχει επισκεπτεί ο κάθε αλγόριθμος. Επειδή μπορούμε να ελέγχουμε αν ένας κόμβος είναι στο σετ σε O(1), αυτή η λύση θα είναι η καλύτερη δυνατή.