Τεχνητή Νοημοσύνη
Εργασία 1

Κωνσταντίνος Χαϊδεμένος
*sdi*2200262

Πρόβλημα 2

Έχουμε ένα δέντρο αναζήτησης με τα εξής χαρακτηριστικά:

- Παράγοντα διακλάδωσης: 3
- Στόχος βρίσκεται σε βάθος: 4
- Οι αλγόριθμοι εκτελούν πλήρη αναζήτηση

Αναζήτηση πρώτα κατά πλάτος (BFS)

min αριθμός κόμβων:

Ο BFS εξετάζει όλους τους κόμβους μέχρι το επίπεδο 4 πριν προχωρήσει σε αυτό για την αναζήτηση του στόχου και βρίσκει στον πρώτο κόμβο του 4ου επιπέδου τον στόχο...

- Επίπεδο 0: *root*
- Επίπεδο 1: 3 κόμβοι
- Επίπεδο 2: $3^2 = 9$ κόμβοι
- Επίπεδο 3: $3^3 = 27$ κόμβοι
- Επίπεδο 4: 1 κόμβος
- Συνολικά: $1 + 3 + 9 + 27 + 1 = 41$ κόμβοι

max αριθμός κόμβων:

Ο BFS εξετάζει όλους τους κόμβους μέχρι το επίπεδο 4 και βρίσκει στον τελευταίο κόμβο του 4ου επιπέδου τον στόχο...

- Επίπεδο 0: $root$
- Επίπεδο 1: 3 κόμβοι
- Επίπεδο 2: $3^2 = 9$ κόμβοι
- Επίπεδο 3: $3^3 = 27$ κόμβοι
- Επίπεδο 4: $3^4 = 81$ κόμβοι
- Συνολικά: $1 + 3 + 9 + 27 + 81 = 121$ κόμβοι

Αναζήτηση πρώτα κατά βάθος (DFS)

Υποθέτουμε οτι η αναζήτηση πρώτα κατά βάθος ξεκινάει από τα αριστερά στο συγκεκριμένο δέντρο

min αριθμός κόμβων:

Ο DFS μπορεί να βρεί τον στόχο έχοντας εξετάσει μόνο 5 καταστάσεις εάν ο στόχος βρίσκεται στο αριστερότερο κόμβο του 4ου επιπέδου...

max αριθμός κόμβων:

Στη χειρότερη περίπτωση που ο στόχος βρίσκεται στον δεξιότερο κόμβο του 4ου επιπέδου ο DFS θα πρέπει να εξετάσει όλους τους κόμβους του δέντρου μέχρι να φτάσει σε αυτόν... Δηλαδή 121 κόμβους

Αναζήτηση με επαναληπτική εκβάθυνση (IDS)

Ο αλγόριθμος αυτός χάνει σε σχέση με τους άλλους δύο στο συγκεκριμένο πρόβλημα λόγω της επαναληπτικότητάς του.
Θα δούμε πως αφού κάθε φορά που αυξάνεται το *depthlimit* πρέπει να ξεκινήσει από τη ρίζα ο συνολικός αριθμός κόμβων που θα εξετάσει θα είναι πολύ μεγαλύτερος...

min αριθμός κόμβων:

Έστω οτι ο στόχος βρίσκεται στον αριστερότερο κόμβο του 4ου επιπέδου...

- 1η επανάληψη: $depth\ limit = 0$, $root$
- 2η επανάληψη: $depth\ limit = 1$, $root + 3 = 4$
- 3η επανάληψη: $depth\ limit = 2$, $root + 3 + 9 = 13$
- 4η επανάληψη: $depth\ limit = 3$, $root + 3 + 9 + 27 = 40$
- 5η επανάληψη: $depth\ limit = 4$, $root + 3 + 9 + 27 + 1 = 41$
- Συνολικά: $1 + 4 + 13 + 40 + 41 = 99$ κόμβοι!
  58 πάνω από τον 2ο καλύτερο (BFS)!!!!

max αριθμός κόμβων:

Ο μέγιστος αριθμός κόμβος θα συναντηθεί όταν χρησιμοποιήσουμε τον IDS στην περίπτωση που ο στόχος ειναι στο δεξιότερο κόμβο του 4ου επιπέδου...
Η μόνη διαφορά με πριν είναι οτί στην τελευταία επανάληψη θα μετρήσει όλους τους κόμβους φύλα.

- Συνολικά: $1 + 4 + 13 + 40 + 121 = 179$ κόμβοιιιιιιι! (πολλοί)

Συμπέρασμα
Αν συγκρίνουμε το $average$ των min, max των αλγορίθμων μπορούμε εύκολα να συμπεράνουμε πως για το συγκεκριμένο πρόβλημα ο πιό αποδοτικός αλγόριθμος θα είναι ο DFS.

Κάτι το οποίο λογικά βγάζει νόημα καθώς ο παράγοντας διακλάδωσης του δέντρου είναι ομοιόμορφος άρα το πλάτος αυξάνεται εκθέτικα σε σχέση με το βάθος που αυξάνεται γραμμικά...

- BFS: $avg(min, max) = 81$
- DFS: $avg(min, max) = \mathbf{63}$
- IDS: $avg(min, max) = 139!!!!!!!!!!$

Πρόβλημα 3

Ο αλγόριθμος A^* θα χρησιμοποιεί την εξής συνάρτηση:

$$f(n) = g(n) + h(n)$$

όπου $g(n)$ είναι το κόστος από την αφετηρία μέχρι τον κόμβο n και $h(n)$ είναι:

$$h(n, G) = \frac{ManhattanDistance(n, G)}{2}$$

όπου G ο στόχος

Λαμβάνουμε ως δεδομένο πως όταν οι κινήσεις που μπορεί να κάνει ο αλγόριθμος είναι ισότιμες θα διαλέγει πάντα να κινηθεί πρός τον αριστερότερο κόμβο.

Η διαδρομή αναζήτησης που θα ακολουθήσει ο A^* καθώς το ρομπότ ψάχνει τον στόχο στην $2D$ αναπαράσταση της πόλης φαίνεται στο παρακάτω σχήμα που μας πήρε καμία ώρα να το φτίαξομυε...

1) $S|$
2) $R|$
3) $R|\ \ |R-R-R|$
4) $R|\_|R \qquad R|-R-R-H-H|$
5) $R|$
6) $R|$
7) $R|$
8) $H|$
9) $R|$
10) $G|$

Το συνολικό κόστος διαδρομής είναι 15.5 ευρά.
Το σύνολο των κόμβων που θα επισκεφτεί το ρομπότ είναι 19.
Η σειρά της εξόδου των κόμβων από το $frontier$ φαίνεται στο σχήμα.

Οι ενέργειες του ρομπότ ήταν με τη σειρά:

κάτω ($x3$) - δεξιά - πάνω - δεξιά ($x3$) - κάτω - δεξιά ($x4$) - κάτω ($x6$) - γκόλ (απλά τα πράγματα)

Δύο νέες ευρετικές συναρτήσεις για τον A^* του συγκεκριμένου προβλήματος είναι οι εξής:

- Ευκλείδια Απόσταση - $h(n, G) = \sqrt{(G_x - n_x)^2 + (G_y - n_y)^2}$
  Είναι παραδεκτή γιατί δεν υπερβαίνει το πραγματικό κόστος.

- Κόστος του Πιο Άμεσου Δρόμου - $h(n, G) = |G_x - n_x| + |G_y - n_y|$
  Είναι παραδεκτή αφού δεν λαμβάνει υπόψη τα εμπόδια.

Πρόβλημα 4

Το περίπλοκο κομμάτι της αμφίδρομης αναζήτησης είναι το σημείο συνάντησης μεταξύ των αλγορίθμων.

Για αυτό τον λόγο θα κάνουμε την εξής παραδοχή:

Ο σκοπός του αλγορίθμου που ξεκινάει από την αρχική κατάσταση θα είναι να συναντήσει τον αλγόριθμο που ξεκινάει από την κατάσταση στόχου οπότε μπορούμε να θεωρήσουμε νέα κατάσταση στόχου του καθενός από το ζευγάρι το σημείο συνάντησης.

Θα αναλύσουμε τις τέσσερεις περιπτώσεις αμφίδρομης αναζήτησης ξεχωριστά:

$\underline{BFS + DLS}$

Γνωρίζουμε (προφανώς) από θεωρία (Διαφάνειες Ενότητας 3 σελίδα 35) πως ο BFS από την αρχική κατάσταση είναι πλήρης και βέλτιστος. Στη συγκεκριμένη περίπτωση, είναι δεδομένο οτι κάποια στιγμή θα συναντήσει τον αλγόριθμο DLS επομένως παραμένει πλήρης και βέλτιστος.

Ο DLS, σε γενικότερη εικόνα προβλήματος, μπορεί να μην είναι πλήρης εφόσον το $depth - limit$ τεθεί πολύ χαμηλά. Αυτό εξαρτάται από το βάθος στο οποίο βρίσκεται η κατάσταση στόχος.
Στη συγκεκριμένη περίπτωση, είναι αυτονόητο οτι κάποια στιγμή θα συναντηθεί με τον BFS επομένως είναι πλήρης. Ωστόσο, παραμένει μη βέλτιστη λύση, όπως και πρόγονός του ο DFS, κάτι το οποίο το γνωρίζουμε πολύ(!) καλά από θεωρία (Διαφάνειες Ενότητας 3 σελίδα 52).

Συνολικά αυτό το ζευγάρι αλγορίθμων είναι πλήρες αλλά δεν αποτελεί βέλτιστη λύση.

$IDS + DLS$

Ο αλγόριθμος IDS είναι πλήρης και βέλτιστος (εφόσον ισχύουν οι προϋποθέσεις του BFS και έχουμε πεπερασμένο χώρο αναζήτησης) ως γνωστόν από θεωρία (Διαφάνειες Ενότητας 2 σελίδα 7 από το τέλος). Προφανώς οι ιδιότητές του διατηρούνται και για την επίλυση του συγκεκριμένου προβλήματος.

Όπως είπαμε και για το προηγούμενο ζευγάρι ο DLS στο συγκεκριμένο πρόβλημα είναι πλήρης αλλά μη βέλτιστος.

Συνεπώς το ζευγάρι αυτό είναι πλήρες αλλά μη βέλτιστο.

$A^* + DLS$

Ο αλγόριθμος A^* είναι πλήρες και βέλτιστος και από όσους έχουμε δεί μέχρι στιγμής είναι και ο καλύτερος (μαντεύω πως είναι και Παναθηναϊκός). Αυτά γνωστά από θεωρία (Διαφάνειες Ενότητας 4 σελίδες 19 και 20). Στο συγκεκριμένο πρόβλημα παραμένει πλήρης και βέλτιστος.

Πάλι έχουμε DLS από κάτω... δεν αλλάζει κάτι που έχουμε A^* από πάνω αφού η ιδιότητά του να είναι μη-βέλτιστος είναι στη φύση του αλγορίθμου γενικότερα.

Στο σημερινό επεισόδιο $algorithm\ island$ το ζευγάρι A^* και DLS είναι πλήρες αλλά... αποδείχτηκε μη-βέλτιστο!

$A^* + A^*$

Εδώ έχουμε διπλό A^* ζευγάρι. Οι δύο αλγόριθμοι εφόσον έχουν την ίδια παραδεκτή ευρετική συνάρτηση θα αποτελέσουν το μόνο πλήρες αλλά και βέλτιστο ζευγάρι από τα 4.
Λογικό αφού και ο αλγόριθμος που ξεκινάει από την αρχική κατάσταση αλλά και αυτός που ξεκινάει από την κατάσταση στόχος έχουν και τις δύο ιδιότητες.

Έλεγχος σημείου συνάντησης

Ο έλεγχος σε όλες τις από πάνω περιπτώσεις ζευγαριών θα γίνει με καταγραφή των καταστάσεων τις οποίες επισκέπτονται τόσο στην "άνοδο" όσο και στην "κάθοδο".
Σε κάθε βήμα θα γίνεται έλεγχος των συνόλων $visited-up$ και $visited-down$ και την στιγμή που βρεθεί κοινό σημείο τότε αυτή η κατάσταση θα αποτελεί και το σημείο συνάντησης των δύο αλγορίθμων.