

AM:1115201800009 ΤΥΠΑΛΔΟΣ-ΠΑΥΛΟΣ ΑΠΟΣΤΟΛΑΤΟΣ

2)

Το best case είναι το DFS να κάνει μια σειριακή αναζήτηση μέχρι το goal state. πχ σε ένα δένδρο να επεκτείνει πάντα τον αριστερό κόμβο μέχρι να βρει τον στόχο που είναι το πιο αριστερό παιδί του επιπέδου $g-1$. πρέπει δηλαδή να επεκτείνει G κόμβους χωρίς να επεκτείνει σε λάθος μονοπάτι με αποτέλεσμα να μην γυρίσει σε ανώτερο επίπεδο και θα σταματήσει εκεί με πολυπλοκότητα $O(G)$. το worst case είναι η πολυπλοκότητα του DFS το b^d δηλαδή να πάει μέχρι το βάθος d και να είναι το τελευταίο leaf του δένδρου (το πιο δεξιο)

3) η συνάρτηση είναι συνεπής γιατί δε μπορείς να βρω ζευγάρι κόμβου με παιδί του που η διαφορά των ευρετικών τους να είναι μεγαλύτερη από το βάρος της ακμής που τους συνδέει. Από τη θεωρία ξέρουμε ότι μια συνεπής συνάρτηση είναι και παραδεκτή

β) επεκτείνοντας τα παιδιά με αλφαβητική σειρά:

DFS (και με επαναληπτική εκβαθύνση)

o103-b3-b1-b2-b4-o109-o111-o119-o123-o125-r123

BFS

o103-b3-o109-ts-b1-b4-o111-o119-mail-b2-c2-o123-storage-c1-c3-o125-r123

greedy: (παιρνοντας κάθε παιδί μέσα στο fringe με το μικρότερο h)

b3-b1-c2-c1-c3-b2-b4-ts-o109-o119-o123-r123

A^* :

$f(o109)=36(6)$

$f(b3)=21 \leftarrow (1)$

$f(ts)=31(5)$

$f(b1)=21 \leftarrow (2)$

$f(b4)=29$

$f(o109)=42$ απορρίπτεται από το fringe

$f(b2)=29(3)$ (βγαίνει πριν το $\beta4(4)$ λόγω αλφαβητικής σειράς και το h καινούρια f του $\beta4$ απορρίπτεται)

(5)

$f(mail)=40(8)$

(6)

$f(o111)=43$

$f(o119)=39(7)$

$f(storage)=47$

$f(o123)=41(9)$

$f(r123)=41(10)$

4)

a)

χώρος καταστάσεων $s = \{(\text{ονομα δωματιου}, [\text{orders}])\}$ όπου orders είναι λίστα πακέτων που πρέπει να πάρει το ρομπότ από αυτό το δωμάτιο σε κάποιο άλλο. Όπως και στο cornersproblem θέλω να ξαναεπισκεπτομαι δωμάτια όσο δεν έχω τελειώσει την συνολική αναζήτηση αρα το [orders] δημιουργεί και άλλες πιθανές καταστάσεις που δεν είναι closed/visited

β)

Εγώ θέλω το ρομπότ να πηγαίνει πρώτα όπου έχει λιγότερη δουλειά αρα ορίζω ερευνητική

$h[\text{state}] = \text{απόσταση ρομπότ από δωμάτιο} + \text{αθροισμα αποστάσεων που πρέπει να διανύσει για κάθε order}$. είναι παραδεκτή επειδή υπολογίζω ακριβώς τα βήματα που χρειάζεται να κάνει το ρομπότ για να ολοκληρώσει ένα δωμάτιο και σε περίπτωση ισοτιμίας φορτού εργασίας ανάμεσα μεταξύ δωματίων το ρομπότ επιλέγει αυτό που είναι πιο κοντά του γεωγραφικά

5)

οι αλγοριθμοι ειναι πληρης δηλαδη θα βρουν ενα μονοπατι για τον στοχο αν στο graphsearch αν το συνολο κομβων ειναι πεπερασμενο ισχυοντας και αν το συνολο s που θα εξερευνησει ο αλγοριθμος απ'την αφετηρια και το s' του αλγοριθμου απο την αφετηρια εχουν μηδενικη τομη γιατι η καθε υπορουτινα μπορεί να φτασει στον στοχο της μονοδρομια χωρις να συναντησει την αλλη. Οι αλγοριθμοι που εχουν ως υπορουτινα dfs δεν εγγυουνται οτι θα επιστρεψουν το βελτιστο μονοπατι στον στοχο. Απ'την αλλη αν στο $\delta(A^* \text{ με } A^*)$ αν δωθει μια συνεπης και παραδεκτη ευρετικη συναρτηση θα επιστρεψει σιγουρα βελτιστο μονοπατι.

Μπορει να γινει ελεγχος οτι συναντηθηκαν αν ελεγχουμε την τομη των closed/visited δομων τους καθε φορα που κανουν expand νεους κομβους