Project 1 – 1115201900113

*Πρόβλημα 2*

(Εφόσον το βάθος ξεκινάει από 0)

**Σύνολο κόμβων δέντρου** : Άθροισμα για n = 0 μέχρι d b^d

Καλύτερη περίπτωση:

Το goal state είναι στο βάθος g. Εάν είναι ο πρώτος κόμβος από αριστερά, θα γίνει αναζήτηση g κόμβων . Αυτή είναι η καλύτερη περίπτωση στην αναζήτηση κατά βάθος , η εξερεύνηση μόνο g κόμβων.

Χειρότερη περίπτωση:

**Κόμβοι από g και κάτω:** Άθροισμα για n = g μέχρι d b^(d-g)

Εάν είναι ο τελευταίος κόμβος από δεξιά στο βάθος g, θα εξερευνήσουμε όλα τα στοιχεία του γράφου, εκτός από τα φύλλα του g και παρακάτω , δηλαδή **Σύνολο κόμβων δέντρου** – **Κόμβοι από g και κάτω.**

*Πρόβλημα 3*

(α) Η συνάρτηση h **είναι συνεπής** αφού για κάθε κόμβο n,n’ με n’ απόγονο του n ισχύει ο τύπος h(n) <= c(n,a,n’) + h(n’)

Δηλαδή, για κάθε κόμβο n ισχύει ότι το h(n) είναι μικρότερο ή ίσο από το άθροισμα h του παιδιού του + την διαδρομή μέχρι αυτό.

Η συνάρτηση h είναι και παραδεκτή γιατί σε κανέναν κόμβο δεν υπερεκτιμάει το κόστος που χρειαζόμαστε για να φτάσουμε στο goal state

(β)

* Αναζήτηση σε πλάτος:

1. o103
2. b3
3. o109
4. ts
5. b1
6. b4
7. o111
8. o119
9. mail
10. b2
11. c2
12. o123
13. storage
14. c1
15. c3
16. o125
17. r123 goal

* Αναζήτηση σε βάθος:

1. o103
2. b3
3. b1
4. b2
5. b4
6. o109
7. o111
8. o119
9. o123
10. o125
11. r123 goal

* Αναζήτηση κατά βάθος με επαναληπτική εκβάθυνση

1. o103
2. b3
3. o109
4. ts
5. o103
6. b3
7. b1
8. b4
9. o109
10. o111
11. o119
12. ts
13. mail
14. o103
15. b3
16. b1
17. b2
18. c2
19. b4
20. o109
21. o111
22. o119
23. o123
24. storage
25. ts
26. mail
27. o103
28. b3
29. b1
30. b2
31. o109
32. o111
33. o119
34. c2
35. c1
36. c3
37. b4
38. o109
39. o111
40. o119
41. o123
42. storage
43. o109
44. o111
45. o119
46. o123
47. o125
48. r123 goal

* Άπληστη αναζήτηση

1. ο103
2. b3
3. b1
4. c2
5. c1
6. c3 end because the graph have directions.

* A\* with heuretic h

1. o103
2. b3
3. b1
4. c2
5. c1
6. b2
7. b4
8. c3
9. ts
10. o109
11. o119
12. mail
13. o123
14. r123 goal

*Πρόβλημα 4*

Mail -> start state

Goal State -> when we are on Mail and all packets have delivered

Check for delivery position farest from the mail state

*Πρόβλημα 5*

O αλγόριθμος σε καμία από τις υπορουτίνες (α) – (δ) δεν είναι πλήρης. Για να είναι πλήρης όπως είπαμε και στην θεωρία θα πρέπει και οι δύο αναζητήσεις να είναι bfs .

Εξίσου και για την ερώτηση αν είναι βέλτιστοι , ισχύει το ίδιο. Εάν δεν είναι και οι δύο αναζητήσεις bfs δεν ισχύουν οι πολυπλοκότητες που αναφέραμε στο μάθημα και ο αλγόριθμος είναι πολύ πιο αργός από την μονόδρομη αναζήτηση . Μάλιστα , ίσως πέσουμε και σε αδιέξοδο και αδυναμία επίλυσης.

Άρα για να μπορέσει να λειτουργήσει σωστά και αποδοτικά η αμφίδρομη αναζήτηση πρέπει και από τα δύο άκρα η αναζήτηση να γίνεται κατά βάθος.

Στην περίπτωση (α) οι δύο αναζητήσεις θα συναντηθούν, χωρίς την αποδοτικότητα που επιθυμούμε.

Στην περίπτωση (β) θα συνατηθούν σίγουρα μόνο εάν θέσουμε στην DLS σαν βάθος μικρότερο του δέντρου

Στην περίπτωση (γ) το βάθος στον DLS θα πρέπει να είναι μεγάλο και να πλησιάζει το βάθος του δέντρου

Στην περίπτωση (δ) , δεν μπορούμε να είμαστε βέβαιοι ότι θα βρεθούν .