National And Kapodistrian University of Athens
Department of Informatics and Telecommunications

Project #1

Student name: *Stylianos Bitzas (sma1100202)*

Course: *Artificial Intelligence (ΥΣ02)* – Professor: *Manolis Koubarakis*
Due date: *November 5th, 2019*

Πρόβλημα 2

Θεωρήστε ένα πρόβλημα αναζήτησης Π που το λύνουμε με τον αλγόριθμο πρώτα σε βάθος με επαναληπτική εκβάθυνση. Έστω ότι το δένδρο αναζήτησης για το Π είναι πεπερασμένο, έχει βάθος d, έχει παράγοντα διακλάδωσης b, και ο κόμβος με το μικρότερο βάθος που αντιστοιχεί σε κατάσταση στόχου βρίσκεται σε βάθος $g \leq d$. Ποιός είναι ο μικρότερος και ποιος ο μεγαλύτερος αριθμός κόμβων που μπορούν να δημιουργηθούν από τον αλγόριθμο; Να εξηγήσετε με λεπτομέρεια την απάντηση σας

Answer. Έστω $Node_g$ κόμβος στόχου που βρίσκεται σε βάθος g≤d και ψάχνουμε να βρούμε το κ=#κόμβων που δημιουργούνται.

- Αν η ρίζα είναι ο κόμβος $Node_g$ τότε $\kappa = 1$
- Αν ο $Node_g$ δεν είναι ρίζα τότε $\kappa > 1$. Σε αυτή την περίπτωση θα έχουμε ότι στη χειρότερη κατάσταση ο αλγόριθμος θα πρέπει να δημιουργεί όλο το δένδρο(αφού ο κόμβος $Node_g$ θα είναι ο τέρμα δεξία στην περιπτώση που οι απόγονοι δημιουργούνται από αριστερά προς στα δεξιά). Επειδή έχουμε τον dfs με επανάληψη και αύξηση του μέγιστου βάθους κάθε φορά αφού η χειρότερη περίπτωση είναι ο $Node_g$ να βρίσκεται σε βάθος d ο αλγόριθμός θα τρέξει τον dfs d+1 φορές. Η ρίζα θα δημιουργηθεί d+1 φορές, τα b παιδιά της ρίζας θα δημιουργηθούν d φορές (αρα θα δημιουργηθούν db κόμβοι),τα b^2 παιδιά των κόμβων αυτών θα δημιουργηθούν d-1 φορές (άρα θα δημιουργηθούν $(d-1)b^2$ κόμβοι) κ.ο.κ. Οπότε τελικά ο τύπος που θα δίνει τον συνολικό αριθμό κ κόμβων που δημιουργούνται θα είναι

$$\kappa = (d+1) + db + (d-1)b^2 + \cdots + 2b^{d-1} + b^d = \sum_{k=0}^{d} (d-k+1)b^k$$

έχουμε

$$\sum_{k=0}^{d} (d-k+1)b^k = \sum_{k=0}^{d} db^k - \sum_{k=0}^{d} kb^k + \sum_{k=0}^{d} b^k \tag{1}$$

$$\sum_{k=0}^{d} b^k = \frac{1-b^{d+1}}{1-b}, \sum_{k=0}^{d} db^k = \frac{d-db^{d+1}}{1-b} \tag{2}$$

$$\sum_{k=0}^{d} kb^k = \left(\sum_{k=0}^{d} b^k\right)' b = \left(\frac{1-b^{d+1}}{1-b}\right)' b = \cdots = \frac{db^{d+1}-(d+1)b^d+1}{(1-b)^2} \tag{3}$$

Άρα τελικά απο (1),(2),(3) έχουμε

$$\sum_{k=0}^{d}(d-k+1)b^k = \cdots = \frac{b^{d+2}-b(d+2)+d+1}{(1-b)^2}$$

Οπότε έχουμε ότι είναι

$$1 \leq \kappa \leq \frac{b^{d+2}-b(d+2)+d+1}{(1-b)^2}$$

Πρόβλημα 3

Για καθένα από τους αλγόριθμους

- Αναζήτηση πρώτα σε πλάτος
- Αναζήτηση πρώτα σε βάθος
- Αναζήτηση πρώτα σε βάθος με επαναληπτική εκβάθυνση
- Άπληστη αναζήτηση πρώτα στον καλύτερο
- A*

να δώσετε: (α) τον κόμβο στόχου στον οποίο φτάνει πρώτα ο αλγόριθμος, και (β) τη σειρά με την οποία βγαίνουν οι κόμβοι από την λίστα «σύνορο» (fringe). Να υποθέσετε ότι: (α) οι αλγόριθμοι έχουν υλοποιηθεί κατάλληλα ώστε να λειτουργούν σωστά σε αλγόριθμους αναζήτησης που είναι γράφοι και (β) όταν ο αλγόριθμος δεν μπορεί να «διακρίνει» δύο κόμβους τότε επιλέγει με αλφαβητική σειρά.

Answer. Ακριβείς οδηγίες θα ήταν πολύ μεγάλη η απάντηση. Άπλα εκτελώ τον αλγόριθμο graphSearch για κάθε αλγόριθμο(dfs,bfs,ids,gbfs,astar) και χρησιμοποιώντας τις ανάλογες δομές (stack,queue,priorityQueue) προκύπτει το αποτέλεσμα.

- ***DFS*** Goal:G_3 Κόμβοι:S,D,E,G_3
- ***BFS*** Goal:G_1 Κόμβοι:S,A,B,D,G_1
- ***IDS*** Goal:G_1 Κόμβοι:S,D,E,C,B,A,G_1
- ***GBFS*** Goal:G_2 Κόμβοι:S,B,C,G_2
- ***ASTAR*** Goal:G_2 Κόμβοι:S,A,B,D,E,G_2

Πρόβλημα 4

Από τα παραπάνω προκύπτει το ακόλουθο πρόβλημα των Σφακιανών πιτών: Μας δίνεται μια αταξινόμητη στοίβα με n Σφακιανές πίτες. Ποιος είναι ο μικρότερος αριθμός από αναποδογυρίσματα f(n) (εκφρασμένος σαν συνάρτηση του n) που θα πρέπει ποτέ να κάνουμε ώστε να τις ταξινομήσουμε κατά τον επιθυμητό τρόπο; Έχετε να απαντήσετε τα εξής ερωτήματα:

1. Αν n=1, 2, 3 ή 4, ποια είναι η τιμή του f(n);
2. Αποδείξτε ότι για $n \geq 4$, έχουμε $f(n) \geq n$.
3. Αποδείξτε ότι για κάθε $n \geq 1$, έχουμε $f(n) \leq 2n$.
4. Να εκφράσετε με ακρίβεια το παραπάνω πρόβλημα σαν πρόβλημα αναζήτησης. Ποιο είναι το μέγεθος του χώρου αναζήτησης του προβλήματος;

Answer.

1. Με απλή απαρίθμηση εξέταζουμε όλες τις περιπτώσεις που είναι μεταθέσεις n αντικειμένων οπότε για n=1,2,3,4 υπάρχουν 1,2,6,24 αρχικές καταστάσεις στοίβας σφακιανών πιτών που δίνουν τιμή f(n)=0,1,3,4 αντίστοιχα.

2. Έστω ότι έχουμε μία στοίβα με n πίτες(μία μετάθεση n στοιχείων) και $i, j \in \{1, 2, \cdots, n\}$ τότε λέμε ότι τα (i, j) σχηματίζουν ένα κακό ζευγάρι αν βρίσκονται ακριβώς δίπλα στην στοίβα και $|i - j| > 1$. Παρατηρούμε ότι αύτες οι 2 πίτες θα παρεμένουν κακό ζευγάρι μέχρις ότου να μπει η τσιμπίδα στη μέση και να τις χωρίσει. Επιπλεόν η πιατέλα με την κατώ πίτα είναι πάντα κακό ζευγάρι. Άρα καταλήγουμε με αυτόν τον συλλογισμό στο ότι μια στοίβα με κ κακά ζευγάρια χρειάζεται τουλάχιστον κ αναποδογυρίσματα για να ταξινομηθεί σωστά. Τελός αρκεί να δούμε ότι για $n \geq 4$ υπάρχει μία στοίβα με n πίτες που περιέχει n κακά ζευγάρια. Για να φανεί αυτό εξετάζουμε δύο περιπτώσεις, την περίπτωση n άρτιος και n περιττός.

   - Αν **n άρτιος** τότε η παρακάτω στοίβα έχει n κακά ζευγάρια:

   $$(2\,4\,6\,\cdots\,n-2\,n\,1\,3\,5\,\cdots\,n-1)$$

   - Αν **n περιττός** τότε η παρακάτω στοίβα έχει n κακά ζευγάρια:

   $$(1\,3\,5\,\cdots\,n-2\,n\,2\,4\,6\,\cdots\,n-1)$$

3. Χρησιμοποιώντας τον παρακάτω αλγόριθμο έχουμε ότι για $n \geq 2$(προφανως για n=1 ισχύει η ανίσωση) $f(n) \leq 2n$.

   Ο αλγόριθμος είναι ο εξής:

   - Αν n=1 μην κάνεις τίποτα
   - Αν n=2 ταξινόμησε τις πίτες με ένα γύρισμα αν δεν είναι ήδη ταξινομημένες
   - Αν >2 φέρε τη μεγαλύτερη πίτα στον πάτο με το πολύ 2 κινήσεις και αναδρομικά ταξινόμησε τις υπόλοιπες n-1 πίτες.

   Άρα έχω 2 κινήσεις για το πολύ n αναδρομές οπότε $f(n) \leq 2n$.

4. **Πρόβλημα Αναζήτησης**
   - **Καταστάσεις:** μεταθέσεις n στοιχείων.
   - **Αρχική Κατάσταση:** μία τυχαία μετάθεση.
   - **Ενέργειες:** Αλλαγή της σειράς των κ πρώτων στοιχείων για $\kappa \leq n$ (π.χ.για n=5 αλλαγή στα πρώτα 3 στοιχεία $(1\ 3\ 4\ 2\ 5) \rightarrow (4\ 3\ 1\ 2\ 5)$)
   - **Τελική Κατάσταση:** n-ταυτοτική μετάθεση
   - **Μέγεθος Χώρου Καταστάσεων:** n!

Προβλήμα 5

Να ορίσετε τυπικά το παραπάνω πρόβλημα σαν πρόβλημα αναζήτησης. Ποιο είναι το μέγεθος του χώρου αναζήτησης; Ποιος είναι ο μέγιστος παράγοντας διακλάδωσης; Για το πρόβλημα του παραπάνω σχήματος, σε ποιο βάθος βρίσκεται η βέλτιστη λύση; Ποιες ευρετικές συναρτήσεις θα χρησιμοποιούσατε αν λύνατε το πρόβλημα με A*;

Answer. Δεν πρόλαβα να βρω μία καλή λύση να παρουσιασώ.