

1^η ΕΡΓΑΣΙΑ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ – ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΠΑΝΑΓΙΩΤΗΣ ΤΣΙΩΝΗΣ ics21012

ΑΥΓΟΥΣΤΟΣ 2023

Το πρόβλημα του Knuth: ξεκινώντας με τον αριθμό 4, μια ακολουθία πράξεων εύρεσης τετραγωνικής ρίζας, ακέραιου μέρους (floor) και παραγοντικών μπορεί να φτάσει σε οποιονδήποτε επιθυμητό θετικό ακέραιο. Για παράδειγμα, μπορούμε να φτάσουμε στον αριθμό 5 με την παρακάτω ακολουθία πράξεων:

Factorial, factorial, root, root, root, root, root, floor.

Για την επίλυση του προβλήματος, κατασκευάστηκε ένα πρόγραμμα σε Java (Maven Project), το οποίο χρησιμοποιεί τους αλγόριθμους **Αναζήτηση πρώτα σε πλάτος (Breadth First Search)** και **Αναζήτηση με επαναληπτική εκβάθυνση (Iterative Deepening Search)**. Στο πρόγραμμα χρησιμοποιείται η βιβλιοθήκη `math3.special.Gamma` της Apache, για τον υπολογισμό παραγοντικού ακεραίων και πραγματικών αριθμών.

Ως είσοδο, ο χρήστης αρχικά πληκτρολογεί τον αριθμό – στόχο που επιθυμεί και έπειτα τον αλγόριθμο αναζήτησης. Για την εξαγωγή συμπερασμάτων, παρακάτω χρησιμοποιούνται ως είσοδο τυχαίοι αριθμοί, ταξινομημένοι σε αύξουσα σειρά, από το 1 έως και το 1000.

Στην κονσόλα του προγράμματος, στην λύση των αλγορίθμων (εάν βρεθεί), το πρώτο βήμα της ακολουθίας ονομάζεται “start”. Για παράδειγμα, εάν η είσοδος είναι ο αριθμός 4, το αποτέλεσμα είναι [start], ενώ αν η είσοδος είναι ο αριθμός 2, το αποτέλεσμα είναι [start, root]. Όμως για λόγους απλότητας, στον παρακάτω πίνακα παραλείπεται.

Η τετραγωνική ρίζα (root) θα αναγράφεται ως “RO”, το ακέραιο μέρος (floor) ως “FL” και το παραγοντικό (factorial) ως “FA”.

Το άτυπο χρονικό όριο κυμαίνεται στα **30 δευτερόλεπτα**. Η διακοπή της αναζήτησης γίνεται **χειροκίνητα** από τον χρήστη μετά το πέρας του ορίου αυτού. Στις περιπτώσεις που οι αλγόριθμοι ξεπεράσουν το όριο, αναγράφεται ως «**ΥΠΕΡΒΑΣΗ ΟΡΙΟΥ**» και θεωρείται ότι δεν βρήκε λύση.

Είσοδος (στόχος)	Λύση BFS	Χρόνος BFS (ms)	Λύση IDS	Χρόνος IDS (ms)
1	RO, RO, FL	10	RO, RO, FL	11
8	FA, RO, RO, FA, FA, FA, FL, FA, RO, RO, FL	181	FA, RO, FA, RO, RO, FA, FA, RO, RO, RO, FA, FL	17
13	FA, RO, FA, RO, FL, FA, RO, RO, RO, RO, FA, FA, FL	1740	FA, RO, FA, FL, RO, FA, FL, RO, FL, RO, FL, RO, RO, FA, FA, FL	37
21	FA, RO, RO, FA, FA, FA, FL, RO, FA, FA, FL	183	FA, RO, FA, FL, FA, RO, RO, RO, RO, RO, RO, RO, RO, FA, RO, FA, RO, FL	67

26	FA, RO, RO, FA, FA, FL, FA, FA, RO, FL	92	FA, FA, RO, FL, RO, FL, RO, FL, RO, FL RO, FA, RO, RO, FA, FL	36
30	FA, FA, RO, RO, RO, RO, FL	16	FA, FA, RO, RO, RO, RO, FL	12
40	FA, RO, FA, RO, RO, FA, RO, FA, FA, FL	92	FA, RO, FA, RO, RO, FA, RO, FA, FA, FL	14
52	FA, RO, FA, FL, RO, FA, RO, RO, RO, FA, RO, FL	596	FA, RO, FA, FL, RO, FA, RO, RO, RO, FA, RO, FL	16
74	-	ΥΠΕΡΒΑΣΗ ΟΡΙΟΥ	FA, FA, RO, RO, RO, RO, RO, FA, RO, FA, FL, RO, RO, RO, RO, FA, RO, RO, RO, FA, RO, FA, RO, RO, FL	4198
121	-	ΥΠΕΡΒΑΣΗ ΟΡΙΟΥ	FA, RO, FA, RO, RO, FA, FA, FL, RO, RO, FA, RO, RO, RO, RO, FA, FA, RO, RO, FA, FA, RO, RO, RO, FA	9055
200	FA, RO, RO, FA, FA, FA, FL, FA, RO, RO, FL, FA, RO, FL	5537	FA, FA, RO, FL, RO, FL, RO, FL, RO, FL, RO, RO, FA, FA, FA, FL, RO, FL, FA, RO, FL	285
301	-	ΥΠΕΡΒΑΣΗ ΟΡΙΟΥ	-	ΥΠΕΡΒΑΣΗ ΟΡΙΟΥ
422	-	ΥΠΕΡΒΑΣΗ ΟΡΙΟΥ	FA, FA, RO, FL, RO, FL, RO, FL RO, RO, FA, RO, RO, FA, RO, FA, RO, RO, FA, RO, FA, RO, RO, FA, RO, FL	8509
586	-	ΥΠΕΡΒΑΣΗ ΟΡΙΟΥ	FA, RO, FA, FA, RO, RO, RO, RO, RO, RO, RO, RO, RO, FA, FA, FA, FA, FA, FA, RO, RO, FA, RO, FL	2162
699	-	ΥΠΕΡΒΑΣΗ ΟΡΙΟΥ	FA, RO, FA, FL, RO, FA, FL, RO, FL, RO, RO, FA, RO, RO, FA, RO, RO, FA, RO, RO, RO, FA, FL	1134

702	-	ΥΠΕΡΒΑΣΗ ΟΡΙΟΥ	FA, RO, FA, FL, RO, FA, FL, RO, RO, RO, FA, RO, RO, FA, FL, RO, RO, FA, RO, RO, RO, FA, FL	1132
888	-	ΥΠΕΡΒΑΣΗ ΟΡΙΟΥ	FA, RO, FA, RO, RO, FA, FA, FL, RO, FL, RO, RO, FA, FA, FL, RO, FL, RO, RO, FA, RO, FA, FL	1232

Με βάση τον παραπάνω πίνακα αποτελεσμάτων, μπορούμε να εξαγάγουμε τα εξής πορίσματα:

- Ο αλγόριθμος αναζήτησης **πρώτα σε πλάτος (BFS)**, ενώ βρίσκει μια λύση για την πλειοψηφία εισόδου μικρών αριθμών (<100), **όσο αυξάνεται το μέγεθος της εισόδου, τόσο περισσότερο χρόνο χρειάζεται ώστε να παράγει λύση**. Στην πλειοψηφία των εισόδων τριψήφιων αριθμών, ξεπέρασε το άτυπο χρονικό όριο που έχει τεθεί, με αποτέλεσμα να μην καθιστάται ικανό να παράγει μια λύση.
- Ο αλγόριθμος αναζήτησης με **επαναληπτική εκβάθυνση (IDS)**, στην συντριπτική πλειοψηφία των εισόδων, **βρήκε μια λύση για το πρόβλημα σε ταχύτερο χρόνο** σε σχέση με τον BFS. Μάλιστα, **κατάφερε να παράγει λύση για σχεδόν όλες εκείνες τις εισόδους τις οποίες ο αλγόριθμος BFS δεν ήταν σε θέση να λύσει**, μέσα στο χρονικό όριο των 30 δευτερολέπτων. Οι περισσότερες λύσεις βρέθηκαν μέσα σε ένα εύλογο χρονικό διάστημα 1,5 – 2 δευτερολέπτων (περίπου 1500 – 2000 ms).
- Για τις εισόδους τις οποίες **και οι δύο αλγόριθμοι ήταν σε θέση να παράξουν μια λύση** μέσα στο χρονικό όριο, παρατηρείται ότι **για καμία είσοδο, η λύση του αλγορίθμου IDS δεν ήταν «καλύτερη» από την λύση του αλγορίθμου BFS**. Συγκεκριμένα, από τις 9 εισόδους για τις οποίες και οι δύο αλγόριθμοι βρήκαν λύση, ο BFS είχε καλύτερη λύση για τις 5 από αυτές (είσοδοι 8, 13, 21, 26, 200), ενώ για τις υπόλοιπες 4 εισόδους, οι δύο αλγόριθμοι πρόσφεραν την ίδια λύση (1, 30, 40, 52). Με τον όρο «καλύτερη λύση», εννοούμε την λύση η οποία απαιτεί μικρότερο πλήθος βημάτων (πράξεων στην συγκεκριμένη περίπτωση) ώστε να φτάσει στον στόχο.

Εν κατακλείδι, με βάση τις παραπάνω δοκιμασίες, μπορούμε να θεωρήσουμε ότι ο αλγόριθμος επαναληπτικής εκβάθυνσης υπερσχύει σε θέμα ταχύτητας, καθώς επίσης είναι προτιμότερος όταν δεν γνωρίζουμε το βάθος της λύσης και υπάρχουν αυστηρά χρονικά όρια ή περιορισμένη μνήμη. Εάν τα παραπάνω δεν αποτελούν πρόβλημα και ως κύριο μέλημα είναι να βρούμε την καλύτερη λύση, τότε θα προτιμηθεί ο αλγόριθμος πρώτα σε βάθος.

Ο πηγαίος κώδικας καθώς και άλλα αρχεία μπορούν να βρεθούν στο GitHub μου, στον συγκεκριμένο σύνδεσμο: [Ergasia1 TN ptsionis](#)