ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΡΓΑΣΙΑ ΜΑΘΗΜΑΤΟΣ: BLOCKS WORLD AGENT

Φοιτητής: Νικόλαος Νικολαΐδης
(ics23080)

Επιβλέπων καθηγητής: Γιάννης Ρεφανίδης

Εργασία 1

Ανάπτυξη agent σε Python για να λυθεί το πρόβλημα του block world με 4 αλγόριθμους αναζήτησης

Δεκέμβριος, 2024

Περιεχόμενα

1. Εισαγωγή....3
2. Υλοποίηση Προγράμματος....4
2.1. Κλάσεις....4
2.2. Συναρτήσεις....4
2.2.1. Ανάλυση κατάστασης και αρχικοποίηση....4
2.2.2. Διαχείριση αναζήτησης....4
2.2.3. Διαχείριση Frontier....5
2.3. Ευρετική συνάρτηση....5
2.4. Έξοδος αποτελέσματος....6
3. Πειραματικά Αποτελέσματα....6
4. Στιγμιότυπα....11
5. Συμπεράσματα....13

1. Εισαγωγή

Το πρόβλημα Blocks World είναι ένα κλασικό πρόβλημα σχεδιασμού και αναζήτησης στην τεχνητή νοημοσύνη, το οποίο έχει μελετηθεί εκτενώς από τις πρώτες μέρες της έρευνας της τεχνητής νοημοσύνης. Σε αυτό το πρόβλημα, έχουμε ένα σύνολο από μπλοκ που μπορούν να στοιβάζονται σε ένα τραπέζι ή πάνω σε άλλα μπλοκ. Κάθε μπλοκ μπορεί να τοποθετηθεί μόνο πάνω σε ένα άλλο μπλοκ ή στο τραπέζι και μόνο τα μπλοκ που δεν έχουν πάνω τους κανένα μπλοκ (καθαρά μπλοκ) μπορούν να μετακινηθούν.

Ο στόχος είναι η μετατροπή μιας αρχικής διαμόρφωσης των μπλοκ σε μια επιθυμητή διαμόρφωση στόχου χρησιμοποιώντας μια σειρά νόμιμων κινήσεων. Μια νόμιμη κίνηση συνίσταται στη λήψη ενός καθαρού μπλοκ και την τοποθέτησή του είτε στο τραπέζι είτε πάνω σε ένα άλλο καθαρό μπλοκ. Η πρόκληση έγκειται στην εύρεση της βέλτιστης ακολουθίας κινήσεων που θα επιτύχει την κατάσταση-στόχο, ελαχιστοποιώντας ταυτόχρονα τον αριθμό των απαιτούμενων βημάτων.

Σε αυτή την ανάθεση, υλοποιήθηκαν τέσσερις διαφορετικοί αλγόριθμοι αναζήτησης για την επίλυση του προβλήματος Blocks World:

1. **Αναζήτηση πρώτα σε πλάτος (BFS)**: Μια μη ενημερωμένη στρατηγική αναζήτησης που εξερευνά συστηματικά το χώρο καταστάσεων επίπεδο προς επίπεδο, εξασφαλίζοντας το συντομότερο μονοπάτι λύσης όσον αφορά τον αριθμό των κινήσεων.

2. **Αναζήτηση πρώτα σε βάθος (DFS)**: Εξερευνά τον χώρο καταστάσεων πηγαίνοντας όσο το δυνατόν βαθύτερα κατά μήκος κάθε κλάδου πριν από την οπισθοδρόμηση, βρίσκοντας ενδεχομένως μια λύση γρήγορα, αλλά χωρίς να εγγυάται τη βέλτιστη λύση.

3. **Καλύτερη πρώτη αναζήτηση**: Μια ενημερωμένη στρατηγική αναζήτησης που χρησιμοποιεί μια ευρετική συνάρτηση για να εκτιμήσει ποιες καταστάσεις είναι πιο ελπιδοφόρες για να εξερευνηθούν στη συνέχεια, δίνοντας προτεραιότητα στις καταστάσεις που φαίνονται πιο κοντά στο στόχο.

4. **A* Αναζήτηση**: Ένας ενημερωμένος αλγόριθμος αναζήτησης που συνδυάζει το πραγματικό κόστος για την επίτευξη μιας κατάστασης με μια ευρετική εκτίμηση του εναπομείναντος κόστους για την επίτευξη του στόχου, παρέχοντας μια βέλτιστη λύση όταν χρησιμοποιείται με μια αποδεκτή ευρετική.

Σκοπός αυτού του έργου είναι να συγκρίνει και να αναλύσει την απόδοση αυτών των διαφορετικών στρατηγικών αναζήτησης στην επίλυση προβλημάτων Blocks World διαφορετικής πολυπλοκότητας. Εξετάζονται παράγοντες όπως ο χρόνος εκτέλεσης, το μήκος της λύσης και ο αριθμός των καταστάσεων που διερευνήθηκαν για να κατανοήσουμε τα πλεονεκτήματα και τους περιορισμούς κάθε προσέγγισης. Αυτή η ανάλυση θα παράσχει πληροφορίες σχετικά με το ποιοι αλγόριθμοι είναι καταλληλότεροι για διαφορετικές περιπτώσεις προβλημάτων και γιατί ορισμένες προσεγγίσεις μπορεί να αποδίδουν καλύτερα από άλλες σε συγκεκριμένα σενάρια.

2. Υλοποίηση Προγράμματος

2.1. Κλάσεις

Κλάση TreeNode: Αυτή η κλάση αναπαριστά κόμβους στο δέντρο αναζήτησης, αποθηκεύοντας:

- ➔ Τρέχουσα κατάσταση διαμόρφωσης
- ➔ Αναφορά γονικού κόμβου για την ανακατασκευή της διαδρομής
- ➔ Κόστος διαδρομής (g)
- ➔ Ευρετική τιμή (h)
- ➔ Συνολική τιμή της συνάρτησης αξιολόγησης (f)
- ➔ Δράση που οδήγησε σε αυτή την κατάσταση

Κλάση Frontier: Υλοποιεί μια δομή συνδεδεμένης λίστας για τη διαχείριση του συνόρου των ανεξερεύνητων κόμβων, υποστηρίζοντας διαφορετικές στρατηγικές εισαγωγής ανάλογα με τον αλγόριθμο αναζήτησης που χρησιμοποιείται.

2.2. Συναρτήσεις

2.2.1. Ανάλυση κατάστασης και αρχικοποίηση

parse_blocks_file
Input: Αρχείο PDDL με περιγραφή προβλήματος
Process: Εξάγει objects, αρχική κατάσταση και στόχο από το PDDL
Output: objects, αρχική κατάσταση (h), στόχος (goal)

2.2.2. Διαχείριση αναζήτησης

initialize_search
Input: Αρχική κατάσταση, μέθοδος, στόχος
Process: Δημιουργεί και αρχικοποιεί τον root node
Output: Αρχικοποιημένο frontier

search
Input: Μέθοδος αναζήτησης
Process: Επαναληπτική αναζήτηση με χρονικό όριο 60s
Output: Κόμβος λύσης ή None

find_children
Input: Τρέχων κόμβος, μέθοδος
Process: Παράγει διαδοχικές καταστάσεις μετακινώντας κύβους
Output: Boolean (επιτυχία δημιουργίας παιδιών)

2.2.3. Διαχείριση Frontier

add_frontier_front
BFS: Προσθήκη στο τέλος (First In First Out)

add_frontier_back
DFS: Προσθήκη στην αρχή (Last In First Out)

add_frontier_in_order
Best/A*: Προσθήκη με βάση την f-value

2.3. Ευρετική συνάρτηση

Η ευρετική συνάρτηση που υλοποιήθηκε βασίζεται στην ιδέα της τροποποιημένης απόστασης Manhattan. Για κάθε κύβο στην τρέχουσα κατάσταση, υπολογίζεται πόσο "μακριά" βρίσκεται από την επιθυμητή του θέση στην κατάσταση-στόχο. Η υλοποίηση της ευρετικής συνάρτησης αποτελείται από δύο κύρια στοιχεία:

- Η **heuristic**(state, goal) προσθέτει τις επιμέρους αποστάσεις για κάθε κύβο
- Η **manhattan_distance**(pile, index, state, goal) υπολογίζει την απόσταση για έναν συγκεκριμένο κύβο.

Μια σημαντική καινοτομία της υλοποίησης είναι η χρήση αρνητικών τιμών για σωστά τοποθετημένες υποακολουθίες κύβων. Συγκεκριμένα, όταν μια ακολουθία κύβων βρίσκεται στη σωστή σειρά (όπως στην κατάσταση-στόχο), η συνάρτηση επιστρέφει την αρνητική τιμή του μήκους της ακολουθίας. Αυτό δημιουργεί ένα "bonus" που ενθαρρύνει τον αλγόριθμο να διατηρήσει τις σωστά τοποθετημένες στοίβες και να επικεντρωθεί στη διόρθωση των λανθασμένα τοποθετημένων κύβων, οδηγώντας σε πιο αποτελεσματική αναζήτηση.

2.4. Έξοδος αποτελέσματος

Η έξοδος του προγράμματος είναι η ακολουθία των ενεργειών που οδηγούν στη λύση, αν αυτή βρεθεί. Αφού ολοκληρωθεί η διαδικασία αναζήτησης, η λύση (αν βρεθεί) γράφεται σε ένα αρχείο, με όνομα που βασίζεται στο αρχείο εισόδου και στην επιλεγμένη μέθοδο αναζήτησης.

Αυτές οι ενέργειες καθορίζονται με την ανίχνευση του πεδίου state και του πεδίου parent από την κλάση TreeNode. Κάθε κόμβος στο δέντρο αναζήτησης αντιπροσωπεύει μια συγκεκριμένη διαμόρφωση των μπλοκ (την κατάσταση) και κάθε κόμβος έχει επίσης μια αναφορά στον κόμβο-γονέα του. Ο γονικός κόμβος αντιπροσωπεύει την κατάσταση από την οποία προήλθε η τρέχουσα κατάσταση. Ακολουθώντας την αλυσίδα των γονικών κόμβων ξεκινώντας από τον κόμβο του στόχου, το πρόγραμμα μπορεί να ανακατασκευάσει την ακολουθία των ενεργειών που έγιναν για την επίτευξη του στόχου.

Ξεκινώντας από τον κόμβο-στόχο, το πρόγραμμα παρακολουθεί το πεδίο των γονέων, συλλέγοντας την ενέργεια που σχετίζεται με κάθε κόμβο. Αυτές οι ενέργειες αντιπροσωπεύουν τις επιμέρους κινήσεις (όπως "Move(block, origin, target)") που εκτελέστηκαν για να μετασχηματίσουν την κατάσταση των μπλοκ. Μόλις συγκεντρωθούν όλες οι ενέργειες, αντιστρέφονται (δεδομένου ότι το ίχνος ξεκινά από το στόχο και εργάζεται αντίστροφα προς την αρχική κατάσταση) για να παραχθεί η σωστή σειρά από την αρχική κατάσταση στην κατάσταση του στόχου.

Εάν δεν βρεθεί λύση εντός του χρονικού ορίου των 60 δευτερολέπτων, το πρόγραμμα θα βγάλει την ένδειξη "Δεν βρέθηκε λύση".

3. Πειραματικά Αποτελέσματα

Πραγματοποιήθηκε σύγκριση των αλγορίθμων (Breadth-First Search, Depth-First Search, Best-First Search και A* Search) με βάση το χρόνο εύρεσης λύσης (σε δευτερόλεπτα).

Problem	breadth	depth	best	astar
probBLOCKS-4-0.txt	0.00	0.00	0.00	0.00
probBLOCKS-4-1.txt	0.00	0.00	0.00	0.00
probBLOCKS-4-2.txt	0.00	0.00	0.00	0.00
probBLOCKS-5-0.txt	0.02	0.01	0.00	0.00
probBLOCKS-5-1.txt	0.02	0.01	0.00	0.00
probBLOCKS-5-2.txt	0.02	0.01	0.00	0.00
probBLOCKS-6-0.txt	0.20	0.09	0.00	0.00
probBLOCKS-6-1.txt	0.38	0.07	0.00	0.00
probBLOCKS-6-2.txt	0.27	0.03	0.00	0.00

probBLOCKS-7-0.txt	17.09	1.21	0.00	0.00
probBLOCKS-7-1.txt	14.59	1.16	0.00	0.00
probBLOCKS-7-2.txt	16.43	0.03	0.00	0.00
probBLOCKS-8-0.txt	No solution	16.06	0.00	0.00
probBLOCKS-8-1.txt	No solution	11.24	0.00	0.00
probBLOCKS-8-2.txt	No solution	10.38	0.00	0.00
probBLOCKS-9-0.txt	No solution	4.10	0.01	0.01
probBLOCKS-9-1.txt	No solution	No solution	0.01	0.01
probBLOCKS-9-2.txt	No solution	No solution	0.01	0.01
probBLOCKS-10-0.txt	No solution	No solution	0.01	0.01
probBLOCKS-10-1.txt	No solution	No solution	0.01	0.03
probBLOCKS-10-2.txt	No solution	No solution	0.03	0.01
probBLOCKS-11-0.txt	No solution	No solution	0.02	0.04
probBLOCKS-11-1.txt	No solution	No solution	0.02	0.02
probBLOCKS-11-2.txt	No solution	No solution	0.02	0.02
probBLOCKS-12-0.txt	No solution	No solution	0.03	0.05
probBLOCKS-12-1.txt	No solution	No solution	0.03	0.04
probBLOCKS-13-0.txt	No solution	No solution	0.06	0.05
probBLOCKS-13-1.txt	No solution	No solution	0.04	0.06
probBLOCKS-14-0.txt	No solution	No solution	0.07	0.10
probBLOCKS-14-1.txt	No solution	No solution	0.05	0.08
probBLOCKS-15-0.txt	No solution	No solution	0.06	0.15
probBLOCKS-15-1.txt	No solution	No solution	0.11	0.10
probBLOCKS-16-1.txt	No solution	No solution	0.11	0.19

probBLOCKS-16-2.txt	No solution	No solution	0.13	0.19
probBLOCKS-17-0.txt	No solution	No solution	0.14	0.13
probBLOCKS-17-0.txt	No solution	No solution	0.11	0.12
probBLOCKS-18-0.txt	No solution	No solution	0.11	0.13
probBLOCKS-18-1.txt	No solution	No solution	0.24	0.16
probBLOCKS-19-0.txt	No solution	No solution	0.31	0.30
probBLOCKS-19-1.txt	No solution	No solution	0.14	0.11
probBLOCKS-20-0.txt	No solution	No solution	0.29	0.29
probBLOCKS-20-1.txt	No solution	No solution	0.25	0.29
probBLOCKS-25-0.txt	No solution	No solution	0.69	0.76
probBLOCKS-25-1.txt	No solution	No solution	0.92	0.87
probBLOCKS-28-0.txt	No solution	No solution	1.21	1.52
probBLOCKS-28-1.txt	No solution	No solution	1.14	1.31
probBLOCKS-30-0.txt	No solution	No solution	2.00	2.26
probBLOCKS-30-1.txt	No solution	No solution	1.63	1.98
probBLOCKS-35-0.txt	No solution	No solution	No solution	No solution
probBLOCKS-35-1.txt	No solution	No solution	No solution	No solution
probBLOCKS-40-0.txt	No solution	No solution	No solution	No solution
probBLOCKS-40-1.txt	No solution	No solution	No solution	No solution
probBLOCKS-45-0.txt	No solution	No solution	No solution	No solution
probBLOCKS-45-1.txt	No solution	No solution	No solution	No solution
probBLOCKS-50-0.txt	No solution	No solution	No solution	No solution
probBLOCKS-50-1.txt	No solution	No solution	No solution	No solution
probBLOCKS-55-0.txt	No solution	No solution	No solution	No solution

probBLOCKS-55-1.txt	No solution	No solution	No solution	No solution
probBLOCKS-60-0.txt	No solution	No solution	No solution	No solution
probBLOCKS-60-1.txt	No solution	No solution	No solution	No solution

Επίσης, πραγματοποιήθηκε σύγκριση των αλγορίθμων (Breadth-First Search, Depth-First Search, Best-First Search και A* Search) με βάση τους κόμβους που δημιουργήθηκαν για να φτάσουν στην λύση.

Problem	breadth	depth	best	astar
probBLOCKS-4-0.txt	4	4	4	4
probBLOCKS-4-1.txt	6	12	6	6
probBLOCKS-4-2.txt	4	9	4	4
probBLOCKS-5-0.txt	7	103	7	7
probBLOCKS-5-1.txt	6	96	6	6
probBLOCKS-5-2.txt	9	66	9	9
probBLOCKS-6-0.txt	7	736	7	7
probBLOCKS-6-1.txt	6	890	6	6
probBLOCKS-6-2.txt	11	530	11	11
probBLOCKS-7-0.txt	11	2587	11	11
probBLOCKS-7-1.txt	12	2284	12	12
probBLOCKS-7-2.txt	12	371	12	12
probBLOCKS-8-0.txt	No solution	47741	10	10
probBLOCKS-8-1.txt	No solution	74739	12	12
probBLOCKS-9-0.txt	No solution	41562	16	16
probBLOCKS-9-1.txt	No solution	No solution	17	17
probBLOCKS-9-2.txt	No solution	No solution	14	14

probBLOCKS-10-0.txt	No solution	No solution	18	18
probBLOCKS-10-1.txt	No solution	No solution	18	18
probBLOCKS-10-2.txt	No solution	No solution	18	18
probBLOCKS-11-0.txt	No solution	No solution	19	19
probBLOCKS-11-1.txt	No solution	No solution	17	17
probBLOCKS-11-2.txt	No solution	No solution	18	18
probBLOCKS-12-0.txt	No solution	No solution	21	21
probBLOCKS-12-1.txt	No solution	No solution	22	22
probBLOCKS-13-0.txt	No solution	No solution	23	23
probBLOCKS-13-1.txt	No solution	No solution	24	24
probBLOCKS-14-0.txt	No solution	No solution	25	25
probBLOCKS-14-1.txt	No solution	No solution	23	23
probBLOCKS-15-0.txt	No solution	No solution	22	22
probBLOCKS-15-1.txt	No solution	No solution	28	28
probBLOCKS-16-1.txt	No solution	No solution	29	29
probBLOCKS-16-2.txt	No solution	No solution	30	30
probBLOCKS-17-0.txt	No solution	No solution	29	29
probBLOCKS-18-0.txt	No solution	No solution	31	31
probBLOCKS-18-1.txt	No solution	No solution	33	33
probBLOCKS-19-0.txt	No solution	No solution	35	35
probBLOCKS-19-1.txt	No solution	No solution	31	31
probBLOCKS-20-0.txt	No solution	No solution	37	37
probBLOCKS-20-1.txt	No solution	No solution	37	37
probBLOCKS-25-0.txt	No solution	No solution	46	46

probBLOCKS-25-1.txt	No solution	No solution	47	47
probBLOCKS-28-0.txt	No solution	No solution	52	52
probBLOCKS-28-1.txt	No solution	No solution	53	53
probBLOCKS-30-0.txt	No solution	No solution	53	53
probBLOCKS-30-1.txt	No solution	No solution	57	57
probBLOCKS-35-0.txt	No solution	No solution	No solution	No solution
probBLOCKS-35-1.txt	No solution	No solution	No solution	No solution
probBLOCKS-40-0.txt	No solution	No solution	No solution	No solution
probBLOCKS-40-1.txt	No solution	No solution	No solution	No solution
probBLOCKS-45-0.txt	No solution	No solution	No solution	No solution
probBLOCKS-45-1.txt	No solution	No solution	No solution	No solution
probBLOCKS-50-0.txt	No solution	No solution	No solution	No solution
probBLOCKS-50-1.txt	No solution	No solution	No solution	No solution
probBLOCKS-55-0.txt	No solution	No solution	No solution	No solution
probBLOCKS-55-1.txt	No solution	No solution	No solution	No solution
probBLOCKS-60-0.txt	No solution	No solution	No solution	No solution
probBLOCKS-60-1.txt	No solution	No solution	No solution	No solution

Η A* αναζήτηση επιδεικνύει σταθερά ανώτερες επιδόσεις, επιλύοντας προβλήματα ταχύτερα και με λιγότερους κόμβους σε σύγκριση με τις BFS και DFS. Αυτό είναι ιδιαίτερα εμφανές σε μεγαλύτερα σύνολα προβλημάτων (π.χ. probBLOCKS-9 και μετά), όπου τα BFS και DFS είτε αποτυγχάνουν να βρουν λύση είτε παρουσιάζουν εκθετική αύξηση των εξεταζόμενων κόμβων. Η Best-First Search αποδίδει επίσης αξιοσημείωτα καλά, φτάνοντας συχνά την A* όσον αφορά τον χρόνο και τον αριθμό των κόμβων, γεγονός που υποδηλώνει ότι μια ισχυρή ευρετική ενισχύει σημαντικά την αποδοτικότητα της αναζήτησης.

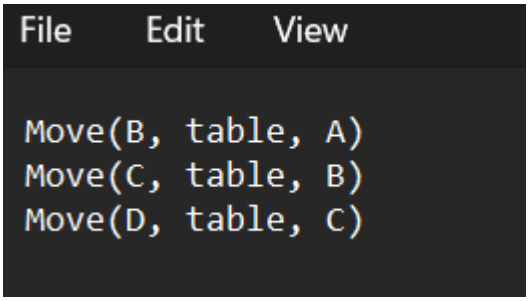
Το DFS παρουσιάζει απότομη αύξηση του αριθμού των παραγόμενων κόμβων καθώς αυξάνεται η πολυπλοκότητα του προβλήματος. Αυτή η τάση αναδεικνύει την έλλειψη καθοδηγούμενης εξερεύνησης στο DFS, που οδηγεί σε υπερβολική επέκταση καταστάσεων

και περιττές αναζητήσεις. Αντίθετα, η BFS, αν και εξαντλητική, κλιμακώνεται ελάχιστα σε χρόνο για μεγαλύτερα προβλήματα (probBLOCKS-8 και πέρα), συχνά σταματώντας χωρίς να δίνει λύση.

Μια εντυπωσιακή παρατήρηση από τον πίνακα δημιουργίας κόμβων είναι ότι το BFS, αν και συχνά αποτυγχάνει στην επίλυση μεγαλύτερων προβλημάτων, επιδεικνύει αποτελεσματικότητα όσον αφορά την ελάχιστη επέκταση κόμβων όταν βρίσκει λύση. Αυτό το μοτίβο είναι ιδιαίτερα αισθητό σε μικρότερους κόσμους μπλοκ (π.χ. probBLOCKS 4-7).

4. Στιγμιότυπα

```
Enter the filename (with or without .txt): probBLOCKS-4-0
Select method: breadth, depth, best, astar: best
Solution found in 0.00 seconds.
Total search time: 0.00 seconds.
Solution actions have been written to 'probBLOCKS-4-0-output-best.txt'.
```




```
Enter the filename (with or without .txt): probBLOCKS-6-0
Select method: breadth, depth, best, astar: depth
Solution found in 0.08 seconds.
Total search time: 0.08 seconds.
Solution actions have been written to 'probBLOCKS-6-0-output-depth.txt'.
```

File Edit View

```
Move(F, E, D)
Move(E, B, table)
Move(B, table, E)
Move(F, D, B)
Move(D, A, F)
Move(A, C, table)
Move(D, F, A)
Move(F, B, D)
Move(B, E, F)
Move(E, table, C)
Move(B, F, E)
Move(F, D, B)
Move(D, A, table)
Move(A, table, D)
Move(F, B, A)
Move(B, E, F)
Move(E, C, table)
Move(B, F, E)
Move(F, A, B)
Move(A, D, F)
Move(D, table, C)
Move(A, F, D)
Move(F, B, A)
Move(B, E, table)
Move(E, table, B)
Move(F, A, E)
Move(A, D, F)
Move(D, C, table)
Move(A, F, D)
Move(F, E, A)
Move(E, B, F)
Move(B, table, C)
Move(E, F, B)
Move(F, A, E)
Move(A, D, table)
Move(D, table, A)
Move(F, E, D)
Move(E, B, F)
Move(B, C, table)
Move(E, F, B)
Move(F, D, E)
Move(D, A, F)
Move(C, table, A)
```

Ln 1, Col 1 11,518 characters

```
Enter the filename (with or without .txt): probBLOCKS-6-0
Select method: breadth, depth, best, astar: breadth
Solution found in 0.17 seconds.
Total search time: 0.17 seconds.
Solution actions have been written to 'probBLOCKS-6-0-output-breadth.txt'.
```

File Edit View

```
Move(D, A, table)
Move(F, E, D)
Move(E, B, F)
Move(A, C, E)
Move(B, table, A)
Move(C, table, B)
```

```
Enter the filename (with or without .txt): probBLOCKS-9-0
Select method: breadth, depth, best, astar: best
Solution found in 0.01 seconds.
Total search time: 0.01 seconds.
Solution actions have been written to 'probBLOCKS-9-0-output-best.txt'.
```

File Edit View

```
Move(F, G, table)
Move(G, E, table)
Move(E, A, table)
Move(A, I, table)
Move(I, D, table)
Move(D, H, table)
Move(H, B, table)
Move(E, table, H)
Move(F, table, E)
Move(I, table, F)
Move(A, table, I)
Move(C, table, A)
Move(B, table, C)
Move(D, table, B)
Move(G, table, D)
```

Άμα ενεργοποιήσουμε τον κώδικα debug print μπορούμε να δούμε τις καταστάσεις της λύσης

```
Select method: breadth, depth, best, astar: astar
Solution found in 0.00 seconds.
Total search time: 0.00 seconds.
Solution actions have been written to 'probBLOCKS-5-0-output-astar.txt'.
[['C', 'D', 'B', 'E', 'A']]
[['A'], ['C', 'D', 'B', 'E']]
[['A'], ['C', 'D', 'B'], ['E']]
[['A', 'B'], ['C', 'D'], ['E']]
[['D'], ['A', 'B'], ['C'], ['E']]
[['D'], ['A', 'B', 'E'], ['C']]
[['D'], ['A', 'B', 'E', 'C']]
```

5. Συμπεράσματα

Τα πειραματικά δεδομένα υπογραμμίζουν τα πλεονεκτήματα και τους περιορισμούς κάθε αλγορίθμου αναζήτησης:

- Η A* επιδεικνύει σταθερά ανώτερες επιδόσεις σε όλες τις περιπτώσεις προβλημάτων, εξισορροπώντας το ευρετικό κόστος και το κόστος διαδρομής για την επίτευξη βέλτιστων λύσεων με ελάχιστη προσπάθεια αναζήτησης.

- Η BFS, αν και δεν είναι τόσο αξιόπιστη για μεγαλύτερα προβλήματα, μπορεί να παράγει λύσεις ελάχιστων βημάτων παρόμοιες με την A* όταν είναι επιτυχής, υποδεικνύοντας τη δυνητική αποτελεσματικότητά της σε συγκεκριμένα σενάρια.

- Η DFS εξερευνά έναν χώρο αναζήτησης πηγαίνοντας όσο το δυνατόν πιο βαθιά σε κάθε κλάδο πριν από την επιστροφή. Αυτό είναι ένα θετικό αποτέλεσμα αφού μπορεί να βρει την λύση ταχυτατα. Ωστόσο, η DFS είναι αναποτελεσματική στην εύρεση της βέλτιστης λύσης, καθώς μπορεί να εξερευνήσει βαθιές διακλαδώσεις που οδηγούν σε μεγάλα μονοπάτια καθώς ψάχνει την λύση.

- Η Best-First Search χρησιμεύει ως βιώσιμη εναλλακτική λύση της A* σε μικρότερα προβλήματα, αλλά μπορεί να παραπαίει σε πολύπλοκα σενάρια που δεν λαμβάνουν υπόψη το κόστος διαδρομής.