

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΡΓΑΣΙΑ #01
Το πρόβλημα του καταχωρητή

ΜΑΡΙΟΣ ΤΣΟΥΛΟΠΟΥΛΟΣ

ΘΕΣΣΑΛΟΝΙΚΗ, 2023

ΠΕΡΙΕΧΟΜΕΝΑ

- 1 Περιγραφή του κώδικα**
- 2 Απαρίθμηση των προβλημάτων**
- 3 Screenshots**
- 4 Σχολιασμός αλγορίθμων**

Περιγραφή του κώδικά

OperationType

Είναι μια κλάση (enum) που περιγράφει κάθε εντολή (increase, decrease,...). Κάθε εντολή έχει 3 method operate, cost, condition που επιστρέφουν

- operate Το αποτέλεσμα της εντολης
- cost Το κοστος της εντολης
- operate Η προϋποθέσεις της εντολης

Node

Η κλάση κόμβος περιέχει

- Τον αριθμό μετα την πράξη (number)
- Την πράξη (operation)
- Το κόστος της πράξης(cost)
- Την αξία της ευρετικής συνάρτησης (h)
- Το βάθος του κόμβου από τη ρίζα (g)
- Το βάρος του κόμβου (f)
- Τον γονέα του κόμβου (parent)

Main

writeResult

Αυτή η μέθοδος χρησιμοποιείται για την εγγραφή αποτελεσμάτων σε ένα αρχείο. Υπολογίζει το συνολικό κόστος προσθέτοντας το κόστος κάθε κόμβου από τη λίστα των κόμβων που περνιέται ως παράμετρος.

isNumeric

Αυτή η μέθοδος ελέγχει εάν μια συμβολοσειρά μπορεί να μετατραπεί σε ακέραιο αριθμό.

isMethod

Αυτή η μέθοδος ελέγχει εάν μια συμβολοσειρά αντιστοιχεί σε ένα από τους αλγόριθμους

main

Η μέθοδος ελέγχει τον αριθμό των παραμέτρων εισόδου και τους μετατρέπει σε διάφορους τύπους μεταβλητών. Έπειτα καλεί την συνάρτηση findPath(method) της κλάσης Algo με παράμετρο τον αλγόριθμο που θα χρησιμοποιηθεί. Τέλος εάν ο αλγόριθμος βρήκε λύση καλεί την συνάρτηση writeResult

Algo

Η κλάση περιέχει 2 μεταβλητές

- start Ο αριθμός που ξεκινάει ο αλγόριθμος
- goal Ο στόχος του αλγόριθμου

Και 4 δομές δεδομένων ο ένας (closed) χρησιμοποιείται για τον έλεγχο ατέρμωνων βρόχων.

Επίσης ένα object TimeUtil που βοηθά στην χρονομέτρηση του αλγορίθμου.

checkClosed

ελέγχει αν η node έχει ίδιο numbers με ένα από τα προηγούμενα

getNode

επιστρέφει μία Node από μία δομή δεδομένων που εξαρτάται από τον τύπο του αλγόριθμου που επιλέχτηκε

isEmpty

ελέγχει αν η δομή δεδομένων είναι άδεια

add

προσθέτει ένα Node σε μια δομή δεδομένων

addAll

προσθέτει ένα Collection απο Node σε μια δομή δεδομένων

findNode

δημιουργεί και έπειτα επιστρέφει τους γειτονικούς κόμβους εφόσον πληρούν τις προϋποθέσεις

getPath

επιστρέφει το μονοπάτι από τον αρχικό κόμβο μέχρι τον κόμβο λύση

isSolution

ελέγχει αν ο αριθμός a είναι λύση

heuristic

συνάρτηση η οποία υπολογίζει το κόστος της πράξης

findPath

Αρχικά δημιουργεί μία μία κόμβο ρίζα που την τοποθετηθεί σε μία δομή δεδομένων. Έπειτα δημιουργεί έναν βρόχο ο οποίος τερματίζεται αν αυτή η δομή δεδομένων είναι άδεια. Μέσα στον βρόχο ελέγχεται αν έχει ξεπεραστεί ο μέγιστος χρόνος εκτέλεσης εάν έχει ξεπεραστεί τότε γράφει αντίστοιχο μήνυμα και επιστρέφει τιμή null αλλιώς ελέγχει αν ο κόμβος είναι λύση εάν πράγματι είναι λύση επιστρέφει το μονοπάτι του κόμβου αλλιώς τοποθετεί στην δομή δεδομένων τους γειτονικούς κόμβους

TimeUtil

μία βοηθητική κλάση που βοηθά τον έλεγχο του χρόνου

AlgorithmMethod

Μία κλάση απαρίθμησης των τεσσάρων αγοριών

Απαρίθμηση των προβλημάτων

αριθμός προβλημάτων	αρχική τιμή	τελική τιμή
1	5	18
2	23	57
3	45	119
4	52	242
5	27	86
6	24	152
7	35	259
8	15	447
9	49	66
10	979	55

Screenshots

1

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mark1\Desktop\test>java -jar Register.jar breadth 5 18 breadth1.txt
Starting solving with BREADTH method
Result found in 5 milliseconds. cost 21 steps 3

C:\Users\mark1\Desktop\test>
```

2

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mark1\Desktop\test>java -jar Register.jar depth 5 18 sdepth1.txt
Starting solving with DEPTH method
Result found in 10 milliseconds. cost 24495550923 steps 987

C:\Users\mark1\Desktop\test>
```

3

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mark1\Desktop\test>java -jar Register.jar best 5 18 best1.txt
Starting solving with BEST method
Result found in 7 milliseconds. cost 11 steps 4

C:\Users\mark1\Desktop\test>
```

4

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mark1\Desktop\test>java -jar Register.jar astar 5 18 sastar1.txt
Starting solving with ASTAR method
Result found in 4 milliseconds. cost 11 steps 4

C:\Users\mark1\Desktop\test>
```

5

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mark1\Desktop\test>java -jar Register.jar breadth 24 57 sbreadth2.txt
Starting solving with BREADTH method
Result found in 9 milliseconds. cost 25 steps 6

C:\Users\mark1\Desktop\test>
```

6

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mark1\Documents\UOM\AI\Register\file>java -jar Register.jar depth 24 57 sdepth2.txt
Starting solving with DEPTH method
Timeout
Not found solution

C:\Users\mark1\Documents\UOM\AI\Register\file>
```

Αξιολόγηση των αλγορίθμων

	breadth			depth			best			astar		
	Χρόνος (millis)	Βήματα	Κόστος	Χρόνος (millis)	Βήματα	Κόστος	Χρόνος (millis)	Βήματα	Κόστος	Χρόνος (millis)	Βήματα	Κόστος
1	5	3	21	10	987	24495 55092 3	7	4	11	4	4	11
2	9	6	25	-	-	-	6	7	27	5	7	27
3	10	5	52	-	-	-	7	16	60	6	11	83
4	11	6	115	1382	48295	17145 91698 088	7	22	138	8	6	116
5	9	5	96	-	-	-	5	9	47	5	9	47
6	16	7	83	-	-	-	7	79	8	6	8	79
7	13	6	81	124	8195	22122 42769 37	7	8	125	7	8	125
8	6	4	173	-	-	-	9	12	231	8	7	226
9	12	6	381	3711	10827 0	37130 72286 748	5	17	34	5	9	56
10	23	9	498	-	-	-	12	304	781	11	10	472