

Το πρόγραμμα δέχεται την είσοδο από ένα αρχείο το οποίο ανοίγει και διαβάζει την αρχική και τελική κατάσταση. Η αρχική και τελική κατάσταση πρέπει να είναι μέσα στο αρχείο αυτό στη μορφή που μας δόθηκε, δηλαδή με τελείες ανάμεσα στους αριθμούς:

0.0.0.0.4.5.1.2.3.7.10.6.8.0.
0.1.4.0.6.0.8.0.0.9.5.0.2.7.

Έχω χρησιμοποιήσει ευρετικές οι οποίες είναι οι εξής

1. int CloseToGoal(vector<int> curr,vector<int> goal)

Παίρνει δύο ορίσματα τα οποία είναι η αρχική και η τελική κατάσταση. Αυτή η ευρετική μόνη της επιλύει μέχρι 14 κύβους σε αποδεκτό χρόνο και για τους 14 αυτούς δημιουργεί 93 κόμβους στο δένδρο αναζήτησης και έχει κόστος συνολικά 13.

2. Άθροισμα ευρετικών CloseToGoal(vector<int> curr,vector<int> goal) και CorrUnder(vector<int> under, vector<int> undergoal)

Η δεύτερη ευρετική που προστίθεται στην πρώτη υπολογίζει αν σε μία δοσμένη κατάσταση είναι σωστά τοποθετημένοι οι κύβοι όχι μόνο από κάτω από κάθε έναν , αλλά και ένας ακόμα πιο κάτω. Αυτό το άθροισμα των ευρετικών δίνει καλύτερα αποτελέσματα από την προηγούμενη για 9 κύβους. Συγκεκριμένα τα αποτελέσματα είναι τα εξής

Για την πρώτη

Initial State = 2.3.0.5.0.7.8.0.6.

Goal State = 0.1.2.0.0.7.0.0.4.

SIZE = 9

Solution found!!!!.

Moved Block number 6

init

Moved Block number 4 to table

Moved Block number 9

Moved Block number 1 to table

Moved Block number 2

Moved Block number 3

Moved Block number 6

Moved Block number 7 to table

Moved Block number 6

nodesExpanded = 449

generatedNodes = 4664

queueSize = 3428

maxQueueSize = 3430

pathCost = 8

Πιέστε ένα πλήκτρο για συνέχεια. . .

Για την δεύτερη μαζί με την πρώτη

Initial State = 2.3.0.5.0.7.8.0.6.

Goal State = 0.1.2.0.0.7.0.0.4.

SIZE = 9

Solution found!!!!.

Moved Block number 6

init

Moved Block number 4 to table

Moved Block number 1 to table

Moved Block number 9

Moved Block number 2

Moved Block number 3

Moved Block number 6

Moved Block number 7 to table

Moved Block number 6

nodesExpanded = 8

generatedNodes = 63

queueSize = 55

maxQueueSize = 56

pathCost = 8

Δηλαδή οι δύο μαζί δημιουργούν σημαντικά λιγότερους κόμβους στο δένδρο, αλλά έχουν ίδιο κόστος μονοπατιού και οι δύο.

Επίσης παρότι πηγαίνουν καλύτερα μαζί για 9 κύβους, δε μπορούν να επιλύσουν παραπάνω.

3. Lasttry(vector<int> cur, vector<int> goal, vector<int> Weight,vector<int> weightgoal)

Αυτή η ευρετική δέχεται την παρούσα και την τελική κατάσταση. Για την παρούσα κατάσταση ,υπολογίζει , για τους λάθος τοποθετημένους κύβους, το βάρος τους, δηλαδή το πόσους κύβους έχουν από πάνω τους, δηλαδή το πόσες κινήσεις ουσιαστικά πρέπει να γίνουν στα σίγουρα. Προσθέτει αυτό το βάρος στο κόστος. Επίσης προσθέτει στο κόστος το βάρος που έχουν οι λάθος κύβοι στην τελική κατάσταση, δηλαδή ουσιαστικά το πόσοι κύβοι θα πρέπει να μετακινηθούν σίγουρα πάνω τους. Επιστρέφει το κόστος.

Αυτή η ευρετική τρέχει σε αποδεκτό χρόνο για μέχρι 27 κύβους δημιουργώντας 2689 κόμβους στο δένδρο και επεκτείνοντας 31. Έχει κόστος μονοπατιού 31.

Για 9 κύβους σε σχέση με τις άλλες δύο έχει κόστος μονοπατιού 5 και δημιουργεί 44 κόμβους.

4. HeuristicNo2(vector<int> state, vector<int> goal)

Αυτή η ευρετική υπολογίζει τον αριθμό των σωστά τοποθετημένων κύβων στην παρούσα κατάσταση και το αφαιρεί από τον αριθμό των λάθος τοποθετημένων κύβων. Ως σωστά τοποθετημένος κύβος ορίζεται ο κύβος που βρίσκεται είτε στο πάτωμα στην τωρινή και την τελική κατάσταση, είτε πάνω σε μία στοίβα κύβων στην οποία είναι και όλοι οι υπόλοιποι σωστοί. Αυτή η ευρετική δουλεύει βέλτιστα για 29 κύβους σε χρόνο περίπου 15 δευτερολέπτων. Ενδεικτικά αποτελέσματα

Initial State = 0.0.0.0.4.5.1.2.3.7.10.6.8.0.9.15.10.13.0.0.14.16.20.0.22.24.21.19.22.

Goal State = 0.1.4.0.6.0.8.0.0.9.5.0.2.7.3.11.12.15.0.13.0.17.0.0.23.25.21.0.24.

SIZE = 29

Solution found!!!!!!.

Moved Block number 20

init

Moved Block number 28 to table

Moved Block number 23 to table

Moved Block number 25

Moved Block number 26

Moved Block number 29

Moved Block number 12 to table

Moved Block number 17

Moved Block number 22

Moved Block number 6 to table

Moved Block number 5

Moved Block number 11

Moved Block number 16

Moved Block number 27

Moved Block number 21 to table

Moved Block number 27

Moved Block number 15

Moved Block number 9 to table

Moved Block number 10

Moved Block number 3

Moved Block number 15

Moved Block number 18

Moved Block number 13

Moved Block number 8 to table

Moved Block number 7

Moved Block number 14

Moved Block number 2

Moved Block number 13

Moved Block number 20

nodesExpanded = 77

generatedNodes = 4533

queueSize = 4456

maxQueueSize = 4457

pathCost = 28

5. Lasttry(curr,goal,Weightcur,Weightgoal)+curr_State.HeuristicNo2(curr,goal)+ curr_State.CloseToGoal(curr, goal);

Πρόκειται για το άθροισμα των δύο παραπάνω ευρετικών το οποίο δουλεύει για περισσότερους κύβους (37) αλλά δουλεύει βέλτιστα για μέχρι 17 κύβους. Ενδεικτικά αποτελέσματα από την εκτέλεση με τους 37 κύβους σε 2 περίπου λεπτά

Initial State = 0.0.0.0.4.5.1.2.3.7.10.6.8.0.9.15.10.13.0.0.14.16.20.0.22.24.21.19.22.26.30.0.
 32.0.0.33.0.
 Goal State = 0.1.4.0.6.0.8.0.0.9.5.0.2.7.3.11.12.15.0.13.0.17.0.0.23.25.21.0.24.29.0.31.34.0.3
 3.28.32.
 SIZE = 37
 Solution found!!!!!!.
 Moved Block number 30
 init
 Moved Block number 12 to table
 Moved Block number 29
 Moved Block number 17
 Moved Block number 6 to table
 Moved Block number 31 to table
 Moved Block number 25
 Moved Block number 22
 Moved Block number 28 to table
 Moved Block number 36
 Moved Block number 33
 Moved Block number 32
 Moved Block number 23 to table
 Moved Block number 25
 Moved Block number 35
 Moved Block number 18
 Moved Block number 16
 Moved Block number 13
 Moved Block number 8 to table
 Moved Block number 5
 Moved Block number 11
 Moved Block number 16
 Moved Block number 27
 Moved Block number 21 to table
 Moved Block number 27
 Moved Block number 15
 Moved Block number 9 to table
 Moved Block number 10
 Moved Block number 7
 Moved Block number 14
 Moved Block number 3
 Moved Block number 15
 Moved Block number 18
 Moved Block number 2
 Moved Block number 13
 Moved Block number 20
 Moved Block number 30
 Moved Block number 26
 Moved Block number 29
 Moved Block number 37
 Moved Block number 30

 nodesExpanded = 40
 generatedNodes = 5234
 queueSize = 5194
 maxQueueSize = 5195
 pathCost = 40
 Πιέστε ένα πλήκτρο για συνέχεια. . .

Κυβοί	CloseToGoal (pathocost)	CloseToGoal+CorrU nder (pathocost)	Lasttry (pathocost)	HeuristicNo2 (pathocost)	LasttryHeuristicNo2+Close ToGoal (pathocost)
14	13	-	15	13	13
15	-	-	16	15	15
8	5	6	5	5	5
17	-	-	20	17	17
29	-	-	-	28	31
30	-	-	-	31	33
37	-	-	-	-	39

Επίσης έχω επισυνάψει μερικά αρχεία κειμένου με ενδεικτικά αποτελέσματα για περισσότερους αριθμούς κύβων για τις διάφορες ευρετικές