ΣΑΚΕΛΛΑΡΗ ΕΛΙΣΑΒΕΤ 1115200600152

Το πρόγραμμα λύνει το πρόβλημα των μαγικών τετραγώνων με τους αλγορίθμους BT, BT+MRV και FC+MRV.Παρατίθεται και ένα αρχείο με συγκεντρωμένα τα αποτελέσματα από τις εκτελέσεις του προγράμματος με τους παραπάνω αλγορίθμους.

Στο αρχείο CSP.cpp υπάρχει η μεταβλητή CONTROL η οποία παίρνει τιμές 0,1,2 και ανάλογα με την τιμή που έχει εκτελείται καθένας από τους αλγορίθμους.
*0: BT
*1: FC+MRV
*2: BT+MRV

Αυτό που κάνει ο MRV και στις δύο περιπτώσεις είναι απλώς να ελέγχει το ποια μεταβλητή έχει μικρότερο πεδίο τιμών και να επιλέγει να δώσει σε αυτήν τιμή.
Στην απλή υπαναχώρηση επιλέγεται η πρώτη μεταβλητή που δεν έχει πάρει κάποια τιμή. Στον πρώιμο έλεγχο, όταν επιλέγεται να δοθεί έγκυρη τιμή σε μία μεταβλητή , από τους γείτονες αυτής της μεταβλητής αφαιρούνται κάποιες πιθανές τιμές. Αυτές είναι οι τιμές που παραβιάζουν τους κανόνες, όταν το άθροισμα της τρέχουσας μεταβλητής και του γείτονά της έχουνε τιμή μεγαλύτερη του επιθυμητού αθροίσματος.
Επιπλέον, όταν από τους γείτονες έχει μείνει μόνο ένα τετράγωνο ακόμα χωρίς να έχει πάρει τιμή, τότε ελέγχω, αν το άθροισμα με τις πιθανές τιμές αυτού του τετραγώνου είναι ακριβώς το επιθυμητό. Αν όχι, τότε σβήνονται αυτές οι τιμές που δεν ικανοποιούν τον περιορισμό.
Στην απλή υπαναχώρηση καλείται η συνάρτηση isSatisfiedWith η οποία κάνει κάποιους ελέγχους , και αν δεν ικανοποιείται κάποιος από αυτούς, επιστρέφει false. Οι έλεγχοι είναι οι εξής :

- Έλεγχος για το αν έχει ξαναδοθεί κάπου αλλού η τιμή.
- Από τους γείτονες του τρέχοντος τετραγώνου, αν έχει μείνει μόνο ένα άδειο τετραγωνάκι, ελέγχω αν σε αυτό μπορεί να μπεί κάποια τιμή έτσι ώστε να προκύπτει τιμή διαφορετική του επιθυμητού αθροίσματος.
- Αν το τετράγωνακι που παει να συμπληρώσει είναι το τελευταίο κάποιας γραμμής, στήλης , ή διαγωνίου, ελέγχω αν το άθροισμα που δίνεται είναι διαφορετικό από το επιθυμητό, οπότε επιστρέφει false.

Το πρόγραμμα συνολικά τρέχει για μέγεθος μέχρι 4.

Δέχεται σαν είσοδο το μέγεθος του πλέγματος των μαγικών τετραγώνων

Ενδεικτικά παραδείγματα εκτέλεσης είναι τα εξής

Ο παρακάτω πίνακας δείχνει ενδεικτικά τον αριθμό των κόμβων που δημιουργούνται ανάλογα με το μέγεθος της εισόδου, και τον αλγόριθμο.

Expanded Nodes

Μέγεθος/αλγόριθμος	BT	BT+MRV	FC+MRV
3	63	33	23
4	3183	148	142

Όπως είναι λογικό ο αλγόριθμος BT+MRV κάνει λιγότερες αναδρομές καθώς δεν επιλέγεται τυχαία η πρώτη ελεύθερη μεταβλητή, αλλά η μεταβλητή με τις περισσότερες έγκυρες τιμές. Αντίστοιχα, ο αλγόριθμος FC+MRV λόγω του ότι μεταβάλλει τα πεδία τιμών, κάνει λιγότερες οπισθοδρομίσεις, καθώς έχει να εξετάσει λιγότερες τιμές.

Consistency checks

Μέγεθος/αλγόριθμος	BT	BT+MRV	FC+MRV
3	522	252	33
4	50792	2232	285

Time

	BT	BT+MRV	FC+MRV
3	0:00.00	0:00.02	0:00.02
4	0:00.82	0:00.34	0:00.35

Επίσης παρατίθεται και ένα αρχείο με ενδεικτικές εκτελέσεις του προγράμματος.

Στον αλγόριθμο BT+MRV δεν μεταβάλλονται τα πεδία των μεταβλητών, απλά ελέγχονται για να αποφασιστεί το ποια μεταβλητή θα πάρει τιμή. Αυτό γίνεται πάλι με τη βοήθεια της συνάρτησης isSatisfiedWith, δίνοντάς της την μετάβλητη που θα ελέγξει και την τιμή για την οποία θα ελέγξει.

Οι μεταβλητές που χρησιμοποιούνται στο πρόγραμμα κρατιούνται σε ένα vector, και τα πεδία ορισμού τους κρατιούνται επίσης σε ένα vector από vectors.

Μια παραδειγματική εκτέλεση με τον αλγόριθμο BT για μέγεθος 3 και 4 είναι η εξής.

```
 ----------BT----------
Complete assignment
Nodes expanded: 63
Consistency checks: 522
Found Solutuion
variable 0_0 has value 2
variable 0_1 has value 7
variable 0_2 has value 6
variable 1_0 has value 9
variable 1_1 has value 5
variable 1_2 has value 1
variable 2_0 has value 4
variable 2_1 has value 3
variable 2_2 has value 8
2 7 6
9 5 1
4 3 8
```

```
----------BT----------
Complete assignment
Nodes expanded: 3183
Consistency checks: 50792
Found Solutuion
variable 0_0 has value 1
variable 0_1 has value 2
variable 0_2 has value 15
variable 0_3 has value 16
variable 1_0 has value 12
variable 1_1 has value 14
variable 1_2 has value 3
variable 1_3 has value 5
variable 2_0 has value 13
variable 2_1 has value 7
variable 2_2 has value 10
variable 2_3 has value 4
variable 3_0 has value 8
variable 3_1 has value 11
variable 3_2 has value 6
variable 3_3 has value 9
1 2 15 16
12 14 3 5
13 7 10 4
8 11 6 9
```

Αλγόριθμος MAC

Ο αλγόριθμος αυτός θα χρησιμοποιούταν για να γίνει συνέπεια τόξου μετά από κάθε ανάθεση κατά την αναζήτηση. Έστω ότι είχαμε το πρόβλημα μεγέθους 3. Τότε αρχικά η μεταβλητή 0_0 θα έπαιρνε την τιμή 1, και τότε θα εφαρμοζόταν ο έλεγχος. Για τις μεταβλητές που σχετίζονται με την μεταβλητή 0_0 θα γινόταν αρχικά έλεγχος για τις μεταβλητές πχ που βρίσκονται στην ίδια γραμμή με την 0_0. Έχοντας κρατήσει το ότι η 0_0 έχει την τιμή 1, ελέγχω ποιες πιθανές τιμές από τις άλλες δύο μεταβλητές της γραμμής θα μπορούσαν να δώσουν το επιθυμητό άθροισμα. Για

παράδειγμα για την τιμή 2 της μεταβλητής 0_1 δεν υπάρχει κάποια τιμή της 0_2 έτσι ώστε αν την προσθέσουμε στο 1+2=3 να δώσει άθροισμα 15. Επομένως επειδή για την 2 δεν υπάρχει κάποια τιμή που να ικανοποιεί τον περιορισμό, αυτή διαγράφεται από το πεδίο της 0_1. Για την τιμή 5 αντίθετα , η τιμή 9 της 0_2 ικανοποιεί τον περιορισμό οπότε παραμένει. Οπότε τώρα το πεδίο της 0_1 θα είναι [5,6,7,8,9]. Πηγαίνοντας στην 0_2 τώρα βλέπουμε ότι η τιμή 2 δεν ικανοποιεί τον περιορισμό , οπότε διαγράφεται. Το ίδιο και η τιμή 3 , κοκ. Οπότε το νέο πεδίο τιμών της 0_2 είναι τώρα το [5,6,7,8,9]. Η ίδια διαδικασία μπορεί να ακολουθηθεί και για τους άλλους γείτονες, στη στήλη και τις διαγωνίους.
Γενικά, αυτό που κάνει αυτός ο αλγόριθμος είναι να αφαιρεί πιο γρήγορα από τα πεδία τιμών των μεταβλητών τις τιμές που μπορεί να οδηγήσουν σε ασυνέπεια.