

# ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ 1<sup>Η</sup>

ΓΙΑΝΝΑΚΟΥΡΟΣ ΠΕΤΡΟΣ 108455 5ο Ετος

## ΑΛΓΟΡΙΘΜΟΣ KRUSKAL

Ο κώδικας υλοποιεί τον αλγόριθμο Kruskal για την εύρεση του Ελάχιστου Δένδρου Εξάπλωσης (Minimum Spanning Tree - MST) σε έναν γράφο.

Αρχικά, ο γράφος ταξινομείται με βάση τα βάρη των ακμών, ώστε να επεξεργαστούμε πρώτα τις ακμές με τα χαμηλότερα βάρη. Στη συνέχεια, δημιουργούνται οι απαραίτητες βοηθητικές δομές δεδομένων. Η δομή `first[v]` αποθηκεύει τον πρώτο κόμβο της λίστας στην οποία ανήκει ο κόμβος `v` και το `my_list[v]` αποθηκεύει το ID της λίστας στην οποία ανήκει. Όλοι οι κόμβοι αρχικά ανήκουν σε ξεχωριστές λίστες, και οι λίστες αυτές καταγράφονται στον πίνακα `components`.

Μετά την αρχικοποίηση, το `TreeEdges[e]` για κάθε ακμή του γράφου ορίζεται σε `false`, δηλώνοντας ότι καμία ακμή δεν ανήκει στο MST αρχικά. Στη συνέχεια, για κάθε ακμή του γράφου, αν οι δύο κόμβοι της ακμής ανήκουν ήδη στην ίδια λίστα, η ακμή παραλείπεται, διότι θα δημιουργούσε κύκλο. Αν ανήκουν σε διαφορετικές συνιστώσες, η ακμή προστίθεται στο MST και οι δύο συνιστώσες συγχωνεύονται. Η συγχώνευση γίνεται με βάση το μέγεθος των συνιστωσών, ώστε η μικρότερη συνιστώσα να ενώνεται με τη μεγαλύτερη για αποδοτικότητα.

Η διαδικασία συνεχίζεται για όλες τις ακμές του γράφου, και τελικά το `TreeEdges` θα περιέχει τις ακμές του MST. Έτσι, ο αλγόριθμος του Kruskal διασφαλίζει την εύρεση του MST με την ελάχιστη δυνατή ποσότητα κόστους, ενώ αποφεύγει τη δημιουργία κύκλων.

## CHECKER KRUSKAL

Αρχικά, ο αλγόριθμος δημιουργεί πίνακες για την αποθήκευση των αποστάσεων (dist) και των προγονικών ακμών (pred) για κάθε κόμβο του γράφου. Όλες οι τιμές της απόστασης αρχικά τίθενται σε -1, .Στη συνέχεια, όλες οι ακμές που δεν ανήκουν στο MST κρύβονται χρησιμοποιώντας τη μέθοδο `G.hide_edge(e)`. Έτσι, όλες οι κρυφές ακμές του γράφου συλλέγονται στη λίστα `h_edges`.

Η επόμενη διαδικασία περιλαμβάνει μια αναζήτηση BFS από έναν τυχαίο κόμβο (τον πρώτο κόμβο του γράφου), για να υπολογιστούν οι αποστάσεις και οι προγονικές ακμές για όλους τους κόμβους του MST. Στη συνέχεια, ο αλγόριθμος ελέγχει όλες τις κρυφές ακμές του γράφου για να διαπιστώσει αν η προσθήκη οποιασδήποτε κρυφής ακμής στο MST παραβιάζει την ελάχιστη απόσταση.

Μετέπειτα, συγκρίνεται η προγονική ακμή του κόμβου με το μεγαλύτερο distance και αν η κρυφή ακμή προσφέρει μικρότερο βάρος από την αντίστοιχη προγονική ακμή, τότε το MST παραβιάζεται και επιστρέφεται false. Αν όλα τα βήματα ολοκληρωθούν χωρίς σφάλματα, ο αλγόριθμος επιστρέφει true, επιβεβαιώνοντας ότι το δέντρο είναι έγκυρο MST.

Τέλος, αφού ολοκληρωθούν όλοι οι έλεγχοι, όλες οι ακμές του γράφου επαναφέρονται στην αρχική τους κατάσταση με τη μέθοδο `G.restore_all_edges()`. Ο αλγόριθμος επιστρέφει το τελικό αποτέλεσμα, είτε true αν το δέντρο είναι έγκυρο, είτε false αν παραβιάζεται κάποια από τις συνθήκες εγκυρότητας του MST.

## ΠΑΡΑΚΑΤΩ ΒΛΕΠΟΥΜΕ SCREENSHOTS ΓΙΑ ΔΙΑΦΟΡΕΤΙΚΕΣ ΕΚΔΟΧΕΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

### ΣΩΣΤΗ ΥΛΟΠΟΙΗΣΗ KRUSKAL=>ΕΠΙΤΥΧΙΑ

```
--- Benchmark για n = 4000, m = 95726 ---
Αθροισμα βαρών MST: 1065043
Αθροισμα βαρών εκτός MST: 478767442
Χρόνος Kruskal (δική μου υλοποίηση): 0.04 sec
Χρόνος Kruskal (LEDA): 0.01 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

--- Benchmark για n = 8000, m = 207452 ---
Αθροισμα βαρών MST: 1936512
Αθροισμα βαρών εκτός MST: 1037379105
Χρόνος Kruskal (δική μου υλοποίηση): 0.17 sec
Χρόνος Kruskal (LEDA): 0.08 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

--- Benchmark για n = 16000, m = 446905 ---
Αθροισμα βαρών MST: 3553627
Αθροισμα βαρών εκτός MST: 2233949809
Χρόνος Kruskal (δική μου υλοποίηση): 0.49 sec
Χρόνος Kruskal (LEDA): 0.22 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

--- Benchmark για πλέγμα 100 x 100 ---
Αθροισμα βαρών MST: 27145701
Αθροισμα βαρών εκτός MST: 72183440
Χρόνος Kruskal (δική μου υλοποίηση): 0.00999999 sec
Χρόνος Kruskal (LEDA): 0.00999999 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

--- Benchmark για πλέγμα 200 x 200 ---
Αθροισμα βαρών MST: 108457733
Αθροισμα βαρών εκτός MST: 290780735
Χρόνος Kruskal (δική μου υλοποίηση): 0.0999999 sec
Χρόνος Kruskal (LEDA): 0.0900002 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

--- Benchmark για πλέγμα 300 x 300 ---
Αθροισμα βαρών MST: 242616997
Αθροισμα βαρών εκτός MST: 655474178
Χρόνος Kruskal (δική μου υλοποίηση): 0.3 sec
Χρόνος Kruskal (LEDA): 0.2 sec
Ο έλεγχος πιστοποίησης: Επιτυχής
```

### ΛΑΝΘΑΣΜΕΝΗ ΥΛΟΠΟΙΗΣΗ=>ΑΠΟΤΥΧΙΑ

```
else
{
    TreeEdges[e] = true;
    if(components[my_list[G.source(e)]].size()<=components[my_list[G.target(e)]].size())
    {
        int previous_id1 = my_list[G.source(e)];
        forall(v,components[my_list[G.source(e)]])
        {
            first[v] = first[G.target(e)];
            // my_list[v] = my_list[G.target(e)];
        }
        components[my_list[G.target(e)]].conc(components[previous_id1]);
    }
    else
    {
        int previous_id2 = my_list[G.target(e)];
        forall(v,components[my_list[G.target(e)]])
        {
            first[v] = first[G.source(e)];
            // my_list[v] = my_list[G.source(e)];
        }
        components[my_list[G.source(e)]].conc(components[previous_id2]);
    }
}

--- Benchmark για n = 16000, m = 446905 ---
Αθροισμα βαρών MST: 2233382862
Αθροισμα βαρών εκτός MST: 4913372
Χρόνος Kruskal (δική μου υλοποίηση): 0.53 sec
Χρόνος Kruskal (LEDA): 0.22 sec
Ο έλεγχος πιστοποίησης: Απέτυχε

--- Benchmark για πλέγμα 100 x 100 ---
Αθροισμα βαρών MST: 96454051
Αθροισμα βαρών εκτός MST: 2723647
Χρόνος Kruskal (δική μου υλοποίηση): 0.00999999 sec
Χρόνος Kruskal (LEDA): 0.00999999 sec
Ο έλεγχος πιστοποίησης: Απέτυχε

--- Benchmark για πλέγμα 200 x 200 ---
Αθροισμα βαρών MST: 389173787
Αθροισμα βαρών εκτός MST: 10571859
Χρόνος Kruskal (δική μου υλοποίηση): 0.11 sec
Χρόνος Kruskal (LEDA): 0.0899999 sec
Ο έλεγχος πιστοποίησης: Απέτυχε

--- Benchmark για πλέγμα 300 x 300 ---
Αθροισμα βαρών MST: 876455155
Αθροισμα βαρών εκτός MST: 23692206
Χρόνος Kruskal (δική μου υλοποίηση): 0.33 sec
Χρόνος Kruskal (LEDA): 0.22 sec
Ο έλεγχος πιστοποίησης: Απέτυχε
```

### ΛΑΝΘΑΣΜΕΝΗ ΥΛΟΠΟΙΗΣΗ=>ΑΠΟΤΥΧΙΑ

```
else
{
    TreeEdges[e] = true;
    if(components[my_list[G.source(e)]].size()<=components[my_list[G.target(e)]].size())
    {
        int previous_id1 = my_list[G.source(e)];
        forall(v,components[my_list[G.source(e)]])
        {
            first[v] = first[G.target(e)];
            my_list[v] = my_list[G.target(e)];
        }
        //components[my_list[G.target(e)]].conc(components[previous_id1]);
    }
    else
    {
        int previous_id2 = my_list[G.target(e)];
        forall(v,components[my_list[G.target(e)]])
        {
            first[v] = first[G.source(e)];
            my_list[v] = my_list[G.source(e)];
        }
        //components[my_list[G.source(e)]].conc(components[previous_id2]);
    }
}

--- Benchmark για n = 4000, m = 95726 ---
Αθροισμα βαρών MST: 479258551
Αθροισμα βαρών εκτός MST: 482844
Χρόνος Kruskal (δική μου υλοποίηση): 0.05 sec
Χρόνος Kruskal (LEDA): 0.03 sec
Ο έλεγχος πιστοποίησης: Απέτυχε

--- Benchmark για n = 8000, m = 207452 ---
Αθροισμα βαρών MST: 1036833762
Αθροισμα βαρών εκτός MST: 444473
Χρόνος Kruskal (δική μου υλοποίηση): 0.22 sec
Χρόνος Kruskal (LEDA): 0.11 sec
Ο έλεγχος πιστοποίησης: Απέτυχε

--- Benchmark για n = 16000, m = 446905 ---
Αθροισμα βαρών MST: 2234481545
Αθροισμα βαρών εκτός MST: 530737
Χρόνος Kruskal (δική μου υλοποίηση): 0.58 sec
Χρόνος Kruskal (LEDA): 0.25 sec
Ο έλεγχος πιστοποίησης: Απέτυχε

--- Benchmark για πλέγμα 100 x 100 ---
Αθροισμα βαρών MST: 92909427
Αθροισμα βαρών εκτός MST: 6484812
Χρόνος Kruskal (δική μου υλοποίηση): 0.00999999 sec
Χρόνος Kruskal (LEDA): 0 sec
Ο έλεγχος πιστοποίησης: Απέτυχε

--- Benchmark για πλέγμα 200 x 200 ---
Αθροισμα βαρών MST: 373445488
Αθροισμα βαρών εκτός MST: 25631382
Χρόνος Kruskal (δική μου υλοποίηση): 0.11 sec
Χρόνος Kruskal (LEDA): 0.0899999 sec
Ο έλεγχος πιστοποίησης: Απέτυχε
```

## ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

```

--- Benchmark για n = 4000, m = 95726 ---
Αθροισμα βαρών MST: 1044605
Αθροισμα βαρών εκτός MST: 476604121
Χρόνος Kruskal (δική μου υλοποίηση): 0.05 sec
Χρόνος Kruskal (LEDA): 0.02 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

```

```

--- Benchmark για n = 8000, m = 207452 ---
Αθροισμα βαρών MST: 1951333
Αθροισμα βαρών εκτός MST: 1034417141
Χρόνος Kruskal (δική μου υλοποίηση): 0.18 sec
Χρόνος Kruskal (LEDA): 0.1 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

```

```

--- Benchmark για n = 16000, m = 446905 ---
Αθροισμα βαρών MST: 3622478
Αθροισμα βαρών εκτός MST: 2237727119
Χρόνος Kruskal (δική μου υλοποίηση): 0.55 sec
Χρόνος Kruskal (LEDA): 0.27 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

```

```
--- Benchmark για πλέγμα 100 x 100 ---
Αθροισμα βαρών MST: 26734652
Αθροισμα βαρών εκτός MST: 71432047
Χρόνος Kruskal (δική μου υλοποίηση): 0.0100002 sec
Χρόνος Kruskal (LEDA): 0.00999975 sec
0 έλεγχος πιστοποίησης: Επιτυχής
```

```

--- Benchmark για πλέγμα 200 x 200 ---
Αθροισμα βαρών MST: 107401680
Αθροισμα βαρών εκτός MST: 290332199
Χρόνος Kruskal (δική μου υλοποίηση): 0.12 sec
Χρόνος Kruskal (LEDA): 0.0999999 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

```

```

--- Benchmark για πλέγμα 300 x 300 ---
Αθροισμα βαρών MST: 243183693
Αθροισμα βαρών εκτός MST: 656854163
Χρόνος Kruskal (δική μου υλοποίηση): 0.34 sec
Χρόνος Kruskal (LEDA): 0.2 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

```

```

--- Benchmark για n = 4000, m = 95726 ---
Άθροισμα βαρών MST: 1063043
Άθροισμα βαρών εκτός MST: 478767442
Χρόνος Kruskal (δική μου υλοποίηση): 0.04 sec
Χρόνος Kruskal (LEDA): 0.01 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

```

```

--- Benchmark για n = 8000, m = 207452 ---
Άθροισμα βαρών MST: 1936512
Άθροισμα βαρών εκτός MST: 1037379105
Χρόνος Kruskal (δική μου υλοποίηση): 0.17 sec
Χρόνος Kruskal (LEDA): 0.08 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

```

```

--- Benchmark για n = 16000, m = 446905 ---
Άθροισμα βαρών MST: 3553627
Άθροισμα βαρών εκτός MST: 2233949809
Χρόνος Kruskal (δική μου υλοποίηση): 0.49 sec
Χρόνος Kruskal (LEDA): 0.22 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

```

```
--- Benchmark για πλέγμα 100 x 100 ---
Άθροισμα βαρών MST: 27145761
Άθροισμα βαρών εκτός MST: 72183440
Χρόνος Kruskal (δική μου υλοποίηση): 0.00999999 sec
Χρόνος Kruskal (LEDA): 0.00999999 sec
Ο έλεγχος πιστοποίησης: Επιτυχής
```

```
--- Benchmark για πλέγμα 200 x 200 ---
Άθροισμα βαρών MST: 108457733
Άθροισμα βαρών εκτός MST: 290780735
Χρόνος Kruskal (δική μου υλοποίηση): 0.0999999 sec
Χρόνος Kruskal (LEDA): 0.0900002 sec
Ο έλεγχος πιστοποίησης: Επιτυχής
```

```
--- Benchmark για πλέγμα 300 x 300 ---
Άθροισμα βαρών MST: 242616997
Άθροισμα βαρών εκτός MST: 655474178
Χρόνος Kruskal (δική μου υλοποίηση): 0.3 sec
Χρόνος Kruskal (LEDA): 0.2 sec
Ο έλεγχος πιστοποίησης: Επιτυχής
```

```
--- Benchmark για n = 4000, m = 95726 ---
Αθροισμα βαρών MST: 1041690
Αθροισμα βαρών εκτός MST: 478623894
Χρόνος Kruskal (δική μου υλοποίηση): 0.04 sec
Χρόνος Kruskal (LEDA): 0.01 sec
Ο έλεγχος πιστοποίησης: Επιτυχής
```

```

--- Benchmark για n = 8000, m = 207452 ---
Αθροισμα βαρών MST: 1943363
Αθροισμα βαρών εκτός MST: 1038944288
Χρόνος Kruskal (δική μου υλοποίηση): 0.18 sec
Χρόνος Kruskal (LEDA): 0.09 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

```

```

--- Benchmark για n = 16000, m = 446905 ---
Αθροισμα βαρών MST: 3600513
Αθροισμα βαρών εκτός MST: 223356861
Χρόνος Kruskal (δική μου υλοποίηση): 0.51 sec
Χρόνος Kruskal (LEDA): 0.24 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

```

```

--- Benchmark για πλέγμα 100 x 100 ---
Αθροισμα βαρών MST: 27342764
Αθροισμα βαρών εκτός MST: 72414223
Χρόνος Kruskal (δική μου υλοποίηση): 0.0100002 sec
Χρόνος Kruskal (LEDA): 0.00999999 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

```

```
--- Benchmark για πλέγμα 200 x 200 ---
Αθροισμα βαρών MST: 107822841
Αθροισμα βαρών εκτός MST: 289299945
Χρόνος Kruskal (δική μου υλοποίηση): 0.1 sec
Χρόνος Kruskal (LEDA): 0.0899997 sec
Ο έλεγχος πιστοποίησης: Επιτυχής
```

```
--- Benchmark για πλέγμα 300 x 300 ---
Αθροισμα βαρών MST: 242410324
Αθροισμα βαρών εκτός MST: 655523080
Χρόνος Kruskal (δική μου υλοποίηση): 0.36 sec
Χρόνος Kruskal (LEDA): 0.24 sec
Ο έλεγχος πιστοποίησης: Επιτυχής
```

```
--- Benchmark για n = 4000, m = 95726 ---
Άθροισμα βαρών MST: 1048288
Άθροισμα βαρών εκτός MST: 479009324
Χρόνος Kruskal (δική μου υλοποίηση): 0.05 sec
Χρόνος Kruskal (LEDA): 0.02 sec
Ο έλεγχος πιστοποίησης: Επιτυχής
```

```
--- Benchmark για n = 8000, m = 207452 ---
Άθροισμα βαρών MST: 1940255
Άθροισμα βαρών εκτός MST: 1035727726
Χρόνος Kruskal (δική μου υλοποίηση): 0.18 sec
Χρόνος Kruskal (LEDA): 0.09 sec
Ο έλεγχος πιστοποίησης: Επιτυχής
```

```

--- Benchmark για n = 16000, m = 446905 ---
Άθροισμα βαρών MST: 3578137
Άθροισμα βαρών εκτός MST: 2233889254
Χρόνος Kruskal (δική μου υλοποίηση): 0.48 sec
Χρόνος Kruskal (LEDA): 0.22 sec
Ο έλεγχος πιστοποίησης: Επιτυχής

```

```

--- Benchmark για πλέγμα 100 x 100 ---
Άθροισμα βαρών MST: 26638211
Άθροισμα βαρών εκτός MST: 71670321
Χρόνος Kruskal (δική μου υλοποίηση): 0.00999999 sec
Χρόνος Kruskal (LEDA): 0.00999999 sec
Ο έλεγχος ισοτιμίας: Επιτυχής

```

```
--- Benchmark για πλέγμα 200 x 200 ---
Άθροισμα βαρών MST: 107633964
Άθροισμα βαρών εκτός MST: 290680212
Χρόνος Kruskal (δική μου υλοποίηση): 0.14 sec
Χρόνος Kruskal (LEDA): 0.12 sec
Ο έλεγχος πιστοποίησης: Επιτυχής
```

```
--- Benchmark για πλέγμα 300 x 300 ---
Άθροισμα βαρών MST: 2422246015
Άθροισμα βαρών εκτός MST: 655749875
Χρόνος Kruskal (δική μου υλοποίηση): 0.35 sec
Χρόνος Kruskal (LEDA): 0.21 sec
Ο έλεγχος πιστοποίησης: Επιτυχής
```

## ΑΝΑΛΥΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

### 1. Τυχαία Συνδεδεμένα Γραφήματα ( $n = 4000, 8000, 16000$ )

Η υλοποίηση του Kruskal που υλοποιήθηκε χειρίζεται με αποτελεσματικότητα μεγάλου μεγέθους τυχαία γραφήματα. Ο χρόνος εκτέλεσης παρουσιάζει αναμενόμενη αύξηση, από 0.04s για 4000 κόμβους έως 0.55s για 16000 κόμβους, διατηρώντας μια γραμμική ή υπολογιστικά αποδεκτή συμπεριφορά ως προς την αύξηση των δεδομένων εισόδου.

Παρόλο που η βιβλιοθήκη LEDA εμφανίζεται σταθερά ταχύτερη — κάτι που ήταν αναμενόμενο, δεδομένων των εξελιγμένων δομών δεδομένων που χρησιμοποιεί— η διαφορά παραμένει μικρή.

Τέλος, η πιστοποίηση των MST με τη βοήθεια του ελέγχου κυκλικότητας (cycle property) ολοκληρώνεται επιτυχώς, επιβεβαιώνοντας τη λογική ορθότητα της κατασκευής του ελάχιστου καλυπτικού δέντρου.

### 2. Πλέγματα (100x100, 200x200, 300x300)

Η απόδοση της υλοποίησης Kruskal σε γράφους πλέγματος είναι ιδιαίτερα ικανοποιητική. Συγκεκριμένα, στο πείραμα με πλέγμα 100x100, η δική μου υλοποίηση ολοκληρώνεται κοντά στο 0.01s, τον ίδιο χρόνο με την Leda .

Αυτό το αποτέλεσμα είναι ενδεικτικό ότι, σε γραμμές τοπολογικά “κανονικές”, όπου οι συνιστώσες αρχικά έχουν προβλέψιμο σχήμα και διασυνδεσιμότητα, η προσέγγιση με συγχωνεύσεις βάσει μεγέθους αποδίδει σταθερά και αποδοτικά.

Η αύξηση του χρόνου μεγεθύνεται αναλογικά με το μέγεθος του πλέγματος, αλλά παραμένει μέσα σε εξαιρετικά χαμηλά επίπεδα ( μέσο όρο 0.34s για 300x300),

αποδεικνύοντας την σταθερότητα και επεκτασιμότητα της υλοποίησης σε πιο πυκνούς και μεγάλους γράφους.

Η κατασκευή των πλεγμάτων γίνεται με τρόπο απόλυτα ελεγχόμενο μέσα από τη συνάρτηση `evaluate_grid_experiment`, όπου δημιουργείται ένας πίνακας διαστάσεων  $r \times c$  και κάθε κόμβος συνδέεται με τους δεξιούς και κάτω γείτονές του.

Όπως και στα υπόλοιπα πειράματα, ο έλεγχος ορθότητας του MST επιβεβαιώνεται επιτυχώς.

### 3. Worst-Case Γράφημα

Η απόδοση της υλοποίησης Kruskal στον χειρότερο περίπτωση για μεγάλους γράφους είναι εξαιρετικά ικανοποιητική και επιδεικνύει εξαιρετική αποδοτικότητα ακόμη και για μεγέθη όπως  $n=4000$ . Η αύξηση του χρόνου απόδοσης με την αύξηση του μεγέθους του γράφου είναι ελάχιστη και παραμένει κάτω από το 1 δευτερόλεπτο, γεγονός που αποδεικνύει την επεκτασιμότητα και τη σταθερότητα της υλοποίησης σε πιο απαιτητικούς και πυκνούς γράφους.

Η κατασκευή των γραφικών δομών γίνεται μέσω της συνάρτησης `evaluate_worst_case_experiment`, η οποία δημιουργεί έναν γράφο με μοτίβο "αλυσίδας/αστέρα", όπου κάθε ακμή αποκτά βάρος ανάλογα με την τοπολογία της. Ο έλεγχος της ορθότητας της λύσης, που γίνεται με την συνάρτηση `checker`, επιβεβαιώνει την εγκυρότητα της υλοποίησης σε κάθε περίπτωση.

WEIGHT(ΑΣΤΕΡΙΟΥ)=10.000 WEIGHT(ΑΛΥΣΙΔΑ)=10

```
--- Benchmark worst-case για n = 2000 ---
Άθροισμα βαρών MST: 29980
Άθροισμα βαρών εκτός MST: 19990000
Χρόνος Kruskal (δική μου υλοποίηση): 0 sec
0 έλεγχος πιστοποίησης: Επιτυχής

--- Benchmark worst-case για n = 4000 ---
Άθροισμα βαρών MST: 49980
Άθροισμα βαρών εκτός MST: 39990000
Χρόνος Kruskal (δική μου υλοποίηση): 0 sec
0 έλεγχος πιστοποίησης: Επιτυχής
```

WEIGHT(ΑΣΤΕΡΙΟΥ)=30.000 WEIGHT(ΑΛΥΣΙΔΑ)=30

```
--- Benchmark worst-case για n = 2000 ---  
Άθροισμα βαρών MST: 89940  
Άθροισμα βαρών εκτός MST: 59970000  
Χρόνος Kruskal (δική μου υλοποίηση): 0 sec  
0 έλεγχος πιστοποίησης: Επιτυχής  
  
--- Benchmark worst-case για n = 4000 ---  
Άθροισμα βαρών MST: 149940  
Άθροισμα βαρών εκτός MST: 119970000  
Χρόνος Kruskal (δική μου υλοποίηση): 0 sec  
0 έλεγχος πιστοποίησης: Επιτυχής
```