

1η Εργαστηριακή Άσκηση

Ονοματεπώνυμο: Γεώργιος Γιάτσος

ΑΜ: 3202

Μάθημα: Θεωρία Γραφημάτων

Διδάσκων: Ιωσήφ Πολενάκης

1. `main()`: Αυτή η συνάρτηση είναι το σημείο εισόδου του προγράμματος. Αρχικά παράγει μια τυχαία ακολουθία βαθμών μήκους 10, όσο δηλαδή επιθυμούμε σε αυτό το παράδειγμα να είναι η τάξη του γραφήματος χρησιμοποιώντας τη συνάρτηση `degree_sequence()`. Στη συνέχεια, η συνάρτηση `graphExists()` ελέγχει αν η ακολουθία είναι γραφική χρησιμοποιώντας τον αλγόριθμο Havel-Hakimi. Εάν η ακολουθία είναι γραφική, παράγει και εμφανίζει ένα γράφημα χρησιμοποιώντας τη συνάρτηση `drawGraph()`. Διαφορετικά, εκτυπώνει ένα κατάλληλο μήνυμα που υποδεικνύει ότι η ακολουθία δεν είναι γραφική.

2. `degree_sequence(n)`: Αυτή η συνάρτηση δέχεται έναν ακέραιο n ως είσοδο και παράγει μια τυχαία ακολουθία βαθμών μήκους n (η τάξη του γραφήματος) χρησιμοποιώντας τον αλγόριθμο Havel-Hakimi. Ο αλγόριθμος ξεκινά με μια λίστα με n μηδενικά και για κάθε επανάληψη επιλέγει έναν τυχαίο αριθμό μεταξύ 0 και $n-1$ και τον προσθέτει στη λίστα. Αυτό συμβαίνει γιατί κάθε κόμβος του γραφήματος μπορεί να έχει βάρος από 0 έως και $n-1$. Έπειτα, μειώνει το n και επαναλαμβάνει τη διαδικασία μέχρι το n να γίνει 0.

3. `graphExists(s)`: Αυτή η συνάρτηση δέχεται ως είσοδο μια λίστα s και ελέγχει αν πρόκειται για μια γραφική ακολουθία βαθμών χρησιμοποιώντας τον αλγόριθμο Havel-Hakimi. Ο αλγόριθμος λειτουργεί ως εξής: ταξινομεί τη λίστα σε μη αύξουσα σειρά, αφαιρεί το πρώτο στοιχείο v και αφαιρεί 1 από τα επόμενα v στοιχεία της λίστας. Εάν κάποιο από τα στοιχεία που προκύπτουν είναι αρνητικό, η ακολουθία δεν είναι γραφική. Εάν όλα τα στοιχεία είναι μη αρνητικά και η λίστα περιέχει μόνο μηδενικά, η ακολουθία είναι γραφική.

4. `drawGraph(sequence)`: Αυτή η συνάρτηση δέχεται ως είσοδο μια ακολουθία λίστας και παράγει ένα γράφημα χρησιμοποιώντας τη βιβλιοθήκη NetworkX. Αρχικά δημιουργεί έναν κενό γράφο και προσθέτει κόμβους με βάρη ίσα με τα στοιχεία της ακολουθίας. Στη συνέχεια προσθέτει ακμές μεταξύ ζευγών κόμβων με βάση την ακολουθία βαθμών, μειώνοντας τους βαθμούς των κόμβων καθώς προστίθενται ακμές. Τέλος, σχεδιάζει το γράφημα χρησιμοποιώντας το Matplotlib.

Παράδειγμα όπου η ακολουθία δεν είναι γραφική:

```
[8]
[8, 1]
[8, 1, 2]
[8, 1, 2, 0]
[8, 1, 2, 0, 9]
[8, 1, 2, 0, 9, 4]
[8, 1, 2, 0, 9, 4, 5]
[8, 1, 2, 0, 9, 4, 5, 6]
[8, 1, 2, 0, 9, 4, 5, 6, 8]
[8, 1, 2, 0, 9, 4, 5, 6, 8, 0]
```

The degree sequence is not graphical!

Παράδειγμα όπου η ακολουθία είναι γραφική και παράγει γράφημα:

```
[2]
[2, 5]
[2, 5, 9]
[2, 5, 9, 2]
[2, 5, 9, 2, 6]
[2, 5, 9, 2, 6, 8]
[2, 5, 9, 2, 6, 8, 4]
[2, 5, 9, 2, 6, 8, 4, 4]
[2, 5, 9, 2, 6, 8, 4, 4, 7]
[2, 5, 9, 2, 6, 8, 4, 4, 7, 5]
```

The degree sequence is graphical using the Havel-Hakimi algorithm!
Here is the Graph!

