

Bilgisayar Uygulamalarında Çizge Kuramı Projesi

Emre Yıldız

25 Ocak 2025

Özet

Bu projede, çizge kuramının temel kavramlarını Python programlama dili kullanarak uyguladık. Proje, çizgelerin oluşturulması, analizi ve manipülasyonu için gerekli temel işlevleri içermektedir. Özellikle, tam çizge, döngü çizge, çark çizge ve n-küp çizge gibi özel çizge türlerinin oluşturulması ve analizi için gerekli algoritmalar implement edilmiştir. Projede herhangi bir harici kütüphane kullanılmamıştır.

İçindekiler

1	Giriş	2
2	Veri Yapıları	2
2.1	Tepe (Vertex) Sınıfı	2
2.2	Çizge (Graph) Sınıfı	2
3	Çizge Girdi/Çıktı İşlemleri	3
3.1	Komşuluk Matrisi Formatı	3
3.2	Komşuluk Listesi Formatı	3
4	Özel Çizge Türleri	3
4.1	Tam Çizge (K_n)	3
4.2	Döngü Çizge (C_n)	3
4.3	Çark Çizge (W_n)	3
4.4	n-Küp Çizge (Q_n)	4

5	Çizge Analizi	4
5.1	Soyutlanmış Tepeler	4
5.2	Asılı Tepeler	4
5.3	İki Parçalı Çizge Kontrolü	4
5.4	Bağlılık Analizi	4
6	Test ve Doğrulama	4
7	Sonuç	5
A	Kaynak Kodları	5

1 Giriş

Çizge kuramı, matematiğin önemli alanlarından biridir ve bilgisayar bilimlerinde geniş uygulama alanına sahiptir. Bu projede, çizge kuramının temel kavramlarını Python programlama dilinde uygulayarak, çizgelerin oluşturulması, analizi ve manipülasyonu için bir yazılım kütüphanesi geliştirdik.

2 Veri Yapıları

Projede kullanılan temel veri yapıları şunlardır:

2.1 Tepe (Vertex) Sınıfı

Tepe sınıfı, çizgedeki her bir düğümü temsil eder. Her tepenin bir etiketi ve komşu tepeler kümesi vardır. Python'un dataclass özelliği kullanılarak implement edilmiştir (bkz. Ek A.1).

2.2 Çizge (Graph) Sınıfı

Çizge sınıfı, tüm çizge operasyonlarını yöneten ana sınıftır. Tepeler ve bunların bağlantılarını içerir. Ayrıca çizge türünü (matris veya liste) belirten bir özelliğe sahiptir (bkz. Ek A.1).

3 Çizge Girdi/Çıktı İşlemleri

Proje iki farklı formatta çizge girişini desteklemektedir (bkz. Ek A.2):

3.1 Komşuluk Matrisi Formatı

Komşuluk matrisi formatında, çizge $n \times n$ boyutunda bir matris ile temsil edilir. Matrisin $[i, j]$ elemanı, i ve j tepeleri arasında kenar varsa 1, yoksa 0 değerini alır.

3.2 Komşuluk Listesi Formatı

Komşuluk listesi formatında, her tepe için o tepenin komşuları listelenir. Bu format, özellikle seyrek çizgeler için daha verimlidir.

4 Özel Çizge Türleri

Proje aşağıdaki özel çizge türlerinin oluşturulmasını desteklemektedir (bkz. Ek A.3):

4.1 Tam Çizge (K_n)

$n \geq 1$ için tam çizge, her tepenin diğer tüm tepelere bağlı olduğu çizgedir. n tepeli tam çizgede $\frac{n(n-1)}{2}$ kenar vardır.

4.2 Döngü Çizge (C_n)

$n \geq 3$ için döngü çizge, n tepenin bir döngü oluşturacak şekilde birbirine bağlandığı çizgedir. Her tepenin derecesi 2'dir.

4.3 Çark Çizge (W_n)

$n \geq 3$ için çark çizge, $n - 1$ tepeli bir döngü çizgeye bir merkez tepenin tüm tepelere bağlanmasıyla oluşur.

4.4 n-Küp Çizge (Q_n)

$n \geq 1$ için n -küp çizge, tepeleri n bitlik ikili sayılarla etiketlenen ve Hamming uzaklığı 1 olan tepelerin birbirine bağlandığı çizgedir.

5 Çizge Analizi

Proje aşağıdaki analiz fonksiyonlarını içermektedir (bkz. Ek A.4):

5.1 Soyutlanmış Tepeler

Hiçbir kenarı olmayan tepelerin bulunması için kullanılır. Bu tepelerin derecesi 0'dır.

5.2 Asılı Tepeler

Sadece bir kenarı olan tepelerin bulunması için kullanılır. Bu tepelerin derecesi 1'dir.

5.3 İki Parçalı Çizge Kontrolü

Çizgenin iki parçalı olup olmadığının kontrolü ve kümelerin bulunması için BFS tabanlı bir renklendirme algoritması kullanılmıştır.

5.4 Bağlılık Analizi

Çizgenin bağlı olup olmadığının kontrolü ve bağlı bileşenlerin bulunması için BFS algoritması kullanılmıştır.

6 Test ve Doğrulama

Proje kapsamlı bir test paketi içermektedir. unittest modülü kullanılarak yazılan testler, tüm fonksiyonların doğru çalıştığını doğrular. Test senaryoları şunları içerir:

- Özel çizge türlerinin doğru oluşturulması
- Çizge özelliklerinin doğru analizi

- Dosya okuma/yazma işlemlerinin doğruluğu

7 Sonuç

Bu projede, çizge kuramının temel kavramları Python programlama dili kullanılarak başarıyla uygulanmıştır. Proje, çizgelerin oluşturulması, analizi ve manipülasyonu için gerekli tüm temel işlevleri içermektedir. Özellikle, farklı çizge türlerinin oluşturulması ve analizi için gerekli algoritmalar başarıyla implement edilmiştir.

Projenin tüm kaynak kodları ve testleri GitHub üzerinde paylaşılmıştır. Proje, çizge kuramı konusunda eğitim ve araştırma amaçlı kullanılabilir.

A Kaynak Kodları

Projenin tam kaynak kodları ayrı bir dokümanda (`code_appendix.tex`) verilmiştir. Bu bölümde kaynak kodların ana bölümleri ve önemli fonksiyonları listelenmiştir.