PROGRAMLAMA LABORATUVARI-II

YAZAR VE İŞBİRLİĞİ ANALİZİ

Aslı Uzar -Mehmet Çağrı Gözen

Bilgisayar Mühendisliği

Kocaeli Üniversitesi

I. ÖZET

Bu proje, akademik bir veri seti kullanılarak yazarlar arasındaki iş birliği ilişkilerini modellemek için bir graf yapısı oluşturmayı ve bu yapı üzerinden çeşitli veri yapısı ve algoritma konseptlerini uygulamayı hedeflemektedir. Öğrenciler, teorik bilgilerini pekiştirirken aynı zamanda pratik becerilerini geliştirecekleri bir süreçten geçeceklerdir.

*a) **Projenin Amaçları:***

1. Graf Veri Yapısını Öğrenmek:

2. Yazarların düğümleri ve yazarlar arasındaki iş birliklerinin kenarları temsil ettiği bir graf yapısı oluşturulacaktır. Bu yapı, graf teorisi ve veri yapıları kavramlarını derinlemesine anlamak için bir temel oluşturacaktır.

3. Veri Yapısı Kavramlarını Uygulamak:

Proje kapsamında, graf üzerinde arama, sıralama, alt ağ oluşturma gibi işlemler yapılacaktır. Bu süreçte öğrencilerin algoritma tasarımı ve uygulama yetenekleri test edilecektir.

4. Gerçek Dünyadan Problem Çözmek:

Akademik iş birliği gibi somut bir problem üzerinden, teorik bilgi pratikle birleştirilerek öğrencilerin problem çözme becerileri güçlendirilecektir.

5. Grafiksel Temsiller ile Çalışmak:

Oluşturulan grafın görselleştirilmesi ve analiz sonuçlarının etkili bir şekilde sunulması beklenmektedir. Bu adım, elde edilen sonuçların daha iyi anlaşılmasını ve yorumlanmasını sağlayacaktır.

*b) **Teknik Kapsam:***

- **Programlama Dili Seçimi:** Proje, belirli bir programlama diline bağlı kalmaksızın gerçekleştirilmekle birlikte, öğrencilerin kendi tercih ettikleri dillerde çalışmalarına olanak tanımaktadır.
- **Hazır Kütüphanelerin Kullanımı:** Veri yapıları ve algoritmalar, öğrenciler tarafından sıfırdan tasarlanmalı ve uygulanmalıdır. Hazır kütüphaneler kullanılmayacaktır.

*c) **Beklenen Çıktılar:***

- Yazarlar arasındaki iş birliği ilişkilerini modelleyen bir graf yapısı.

- Graf üzerinde gerçekleştirilen algoritmaların sonuçları: arama, sıralama, alt ağ oluşturma.
- Grafın etkili bir şekilde görselleştirilmesi ve analizlerin raporlanması.

Bu proje, öğrencilerin hem teorik bilgilerini pekiştirmelerine hem de gerçek bir problemi çözerek uygulama becerilerini geliştirmelerine olanak tanıyan kapsamlı bir öğrenme deneyimi sunmaktadır.

II. GİRİŞ

Akademik iş birliği, bilimsel üretimin önemli bir unsuru olup, yazarlar arasındaki iş birliği ilişkilerinin incelenmesi, bilimsel yayınların daha iyi anlaşılmasını ve çeşitli analizlerin yapılmasını sağlamaktadır. Bu projede, bir akademik veri seti kullanılarak yazarlar arasındaki iş birliği ilişkilerinin bir graf yapısı aracılığıyla modellenmesi ve bu model üzerinden veri yapısı ve algoritma konseptlerinin uygulanması hedeflenmektedir. Proje, hem teorik hem de pratik yönleriyle öğrencilerin analitik düşünme ve yazılım geliştirme becerilerini geliştirmelerine olanak tanıyacaktır.

Projenin temel amacı, verilen bir veri seti üzerinde çalışarak yazarlar arasındaki iş birliği ilişkilerini anlamlı bir şekilde modellemek ve bu model üzerinden algoritmik işlemler gerçekleştirmektir. Veri seti, makale başlıkları, yazarlar ve makalelere ait benzersiz dijital tanımlayıcıları (DOI) içermektedir. Bu veri seti, Python programlama dili kullanılarak analiz edilmekte ve işlenmektedir. Python'un zengin kütüphane desteği sayesinde veri okuma, analiz ve görselleştirme süreçleri etkin bir şekilde gerçekleştirilmektedir.

Projenin veri işleme aşamasında **pandas** kütüphanesi kullanılmaktadır. Pandas, Excel formatındaki veri setinin hızlı ve verimli bir şekilde okunması, işlenmesi ve analiz edilmesi için güçlü araçlar sunmaktadır. Veri setindeki yazar bilgileri işlenerek her yazar için benzersiz bir ID atanmakta ve yazarlar arasındaki iş birlikleri birer kenar (edge) olarak graf modeline eklenmektedir. Aynı zamanda, düğüm ve kenar özellikleri (örneğin, düğüm boyutu, renk, kenar ağırlığı) belirlenerek graf görselleştirme için gerekli altyapı oluşturulmaktadır.

Grafın görselleştirilmesi ve kullanıcıya dinamik bir şekilde sunulması için projede **Flask** ve **Pyvis** kütüphaneleri kullanılmaktadır. Flask, projeye bir web tabanlı arayüz ekleyerek kullanıcı etkileşimini kolaylaştırırken, Pyvis kütüphanesi graf yapılarının görselleştirilmesini ve etkileşimli bir şekilde analiz edilmesini sağlamaktadır. Kullanıcılar, arayüz üzerinden graf üzerindeki düğüm ve kenarları inceleyebilir, belirli algoritmaları çalıştırabilir ve sonuçları görsel olarak takip edebilir.

Bu proje kapsamında oluşturulan graf modelinde, yazarlar düğümler (nodes) olarak temsil edilmekte ve yazarlar arasındaki iş birlikleri kenarlar (edges) aracılığıyla gösterilmektedir. Kenarlar ağırlıklı olup, ağırlıklar iki yazarın kaç ortak makale yazdığını ifade etmektedir. Ayrıca, yazarların iş birliği yoğunlukları düğüm boyutları ve renkleri ile görselleştirilmekte, böylece kullanıcılar ilişkileri daha kolay analiz edebilmektedir.

Projenin genel amacı, yalnızca akademik iş birliklerini modellemek değil, aynı zamanda bu model üzerinden veri yapısı ve algoritmalarla ilgili farklı işlemler gerçekleştirmektir. Proje, kullanıcıya şu işlemleri yapma imkânı sunmaktadır:

- İki yazar arasındaki en kısa yolun bulunması ve grafiksel olarak gösterilmesi.
- Belirli bir yazarın iş birliği yaptığı diğer yazarlarla bir kuyruk oluşturulması.
- Kuyruktan bir ikili arama ağacı (Binary Search Tree) oluşturulması.
- Belirli bir yazarın iş birliği yaptığı diğer yazarlarla ilişkili kısa yolların hesaplanması.
- En çok iş birliği yapan yazarların ve en uzun yolların analiz edilmesi.

Projede kullanılan teknolojiler ve metodolojiler, öğrencilerin yalnızca programlama becerilerini değil, aynı zamanda veri analizi, algoritma geliştirme ve görselleştirme alanlarındaki yetkinliklerini artırmayı amaçlamaktadır. Python'un sağladığı esneklik ve kütüphane desteği sayesinde proje, hem teknik hem de analitik açıdan etkili bir çözüm sunmaktadır. Bu süreçte öğrenciler, somut bir problem üzerinden soyut veri yapısı ve algoritma kavramlarını uygulayarak kapsamlı bir öğrenme deneyimi elde etmektedir.

III. YÖNTEM

Bu projede akademik işbirliklerini modelleyen bir graf yapısı oluşturmak ve kullanıcıların bu graf üzerinde çeşitli işlemleri gerçekleştirebileceği bir analiz platformu geliştirmek için aşağıdaki yöntemler uygulanmıştır. Çalışma, veri işleme, graf modelleme, algoritma tasarımı ve görselleştirme adımlarından oluşmaktadır.

a) Veri İşleme ve Hazırlık

Projenin ilk aşamasında, bir Excel formatında verilen akademik iş birliği veri seti işlenmiştir. Veri işleme için Python programlama dili kullanılmıştır ve veri okuma işlemleri için **pandas** kütüphanesi tercih edilmiştir.

- **Excel'den Veri Çekme:**

- Veri setinde her bir satırda bir makale bulunmaktadır. Makale başlığı (Title), yazarlar (Co-authors) ve DOI (Digital Object Identifier) bilgileri bu satırlardan çekilmiştir.

- **Yazarların Benzersiz Kimliklerle Tanımlanması:**

Yazarlar, düğümler (nodes) olarak modellenmiştir ve her bir yazara benzersiz bir kimlik (ID) atanmıştır. Aynı isimli yazarlar tek bir yazar olarak kabul edilmiştir.

- **Graf Yapısının Oluşturulması:**

Yazarlar arasındaki ortak makale ilişkileri, graf kenarları (edges) ile temsil edilmiştir. Kenar ağırlıkları, iki yazarın ortak makale sayısını göstermektedir. Veri setindeki bilgiler, bir graf veri yapısına dönüştürülerek depolanmıştır.

b) Graf Modelleme

Graf yapısı, akademik iş birliklerini anlamlı bir şekilde görselleştirmek ve analiz etmek için tasarlanmıştır.

- **Düğüm ve Kenar Özellikleri:**
  - **Düğüm Boyutları:** Yazarların yazdığı toplam makale sayısına göre belirlenmiştir. Ortalama makale sayısının %20 üzerinde olan düğümler daha büyük ve koyu renkli, %20 altında olanlar ise daha küçük ve açık renkli olarak gösterilmiştir.
  - **Kenar Ağırlıkları:** İki yazar arasındaki ortak makale sayısına dayalı olarak hesaplanmıştır.
- **Dinamik Etkileşim:**
  - Kullanıcı, graf üzerinde düğümlere tıklayarak yazarın bilgilerini (isim ve makaleler) görebilir.
  - Fare ile graf üzerinde kaydırma (pan) ve yakınlaştırma (zoom in/out) özellikleri eklenmiştir.
- **Görselleştirme:**

Graf, görselleştirme işlemleri için **Pyvis** kütüphanesi kullanılarak oluşturulmuştur. Pyvis, düğüm ve kenarların renk, boyut gibi görsel özelliklerini etkileşimli bir şekilde yönetmeyi sağlamıştır.

c) Algoritma Tasarımı ve Uygulamaları

Proje kapsamında belirli işlemleri gerçekleştirebilmek için farklı algoritmalar tasarlanmış ve uygulanmıştır.

1. **İki Yazar Arasındaki En Kısa Yolun Hesaplanması:**
    a. Kullanıcıdan iki yazarın kimlik bilgileri alınır.
    b. Graf yapısında, Dijkstra algoritması gibi ağırlıklı en kısa yol algoritmaları kullanılarak en kısa yol hesaplanır.
    c. Hesaplama sonucunda, yol üzerindeki düğümler sırasıyla kullanıcıya gösterilir ve adım adım vurgulanır.
2. **Yazarların Kuyruk Yapısında Sıralanması:**
    a. Bir yazarın iş birliği yaptığı diğer yazarlar, düğüm ağırlıklarına (makale sayılarına) göre bir öncelikli kuyrukta sıralanmıştır.
    b. Kuyruğa ekleme ve çıkarma işlemleri dinamik olarak gösterilmiştir.
3. **İkili Arama Ağacı (Binary Search Tree) Oluşturma:**
    a. Kuyrukta bulunan yazarlar bir ikili arama ağacına dönüştürülmüştür.
    b. Kullanıcı, belirli bir yazarı ağacın dışına çıkardığında, ağacın güncel durumu görselleştirilmiştir.
4. **Yazarlar Arasında Alternatif Kısa Yolların Belirlenmesi:**
    a. Kullanıcıdan bir yazar ID'si alınarak, bu yazarın iş birliği yaptığı yazarlar ve dolaylı olarak iş birliği kurduğu diğer yazarlar arasında alternatif kısa yollar hesaplanmıştır.
    b. Süreç boyunca graf üzerinde tablolar güncellenmiş ve kullanıcıya adım adım gösterilmiştir.
5. **İşbirliği Yoğunluğunun Analizi:**
    a. Belirli bir yazarın iş birliği yaptığı toplam yazar sayısı ve graf üzerindeki en iş birliği yoğun yazarı belirlemek için algoritmalar tasarlanmıştır.
6. **En Uzun Yolun Hesaplanması:**
    a. Kullanıcıdan alınan bir yazar kimliği ile başlayan ve tüm düğümleri yalnızca bir kez ziyaret ederek oluşan en uzun yol hesaplanmıştır.

d) Kullanıcı Arayüzü ve Dinamik Sunum

Kullanıcı ile etkileşim için **Flask** web çerçevesi kullanılmıştır.

- **Arayüz Özellikleri:**
    - Kullanıcı, belirli bir istekte bulunmak için yazar kimliklerini arayüze girebilir.
    - Grafiksel sonuçlar, adım adım güncellenen bir panelde gösterilir.
    - İşlem sırasında kullanılan düğümler, farklı renklerle vurgulanır ve süreç dinamik bir şekilde izlenebilir.
- **Sonuçların Gösterimi:**
    - Kullanıcılar, analiz sonuçlarını görsel bir graf üzerinden inceleyebilir.
    - Her işlem sonrası grafik güncellenir ve kullanıcılar işlemlerin detaylarını bir yan panelde takip edebilir.

e) Çıktılar ve Veri Kaydı

Proje boyunca yapılan analizlerin çıktıları ve süreç detayları kaydedilmiştir:

- Her bir işlem sonrası, sonuçlar ve adımlar bir metin dosyasına kaydedilmiştir.
- Graf işlemleri sırasında gerçekleştirilen algoritmaların ara sonuçları görselleştirme ekranında kullanıcıya gösterilmiştir.

Bu yöntemler, projenin hem teorik hem de uygulamalı hedeflerini gerçekleştirmesini sağlamış ve kullanıcıya zengin bir deneyim sunmuştur

IV. DENEYSEL SONUÇLAR

a) Deneyin ve Simülasyonun Tanıtımı

Bu çalışmada, akademik yazarlar arasındaki iş birliği ilişkilerini modellemek ve bu model üzerinde çeşitli veri yapısı ve algoritma konseptlerini uygulamak amacıyla bir graf tabanlı simülasyon geliştirilmiştir. Çalışma, Python programlama dili kullanılarak gerçekleştirilmiş, veri işleme ve görselleştirme için bazı yardımcı kütüphanelerden yararlanılmıştır. Projede kullanılan yöntemler ve araçlar, gerçek dünya problemlerine yönelik çözüm üretme yeteneğini geliştirmeyi ve temel veri yapılarıyla algoritmaları uygulamalı bir şekilde öğrenmeyi amaçlamaktadır.

b) Simülasyon Yazılımı

Simülasyonun yazılım kısmı aşağıdaki araçlar ve yöntemler ile gerçekleştirilmiştir:

- **Python Programlama Dili:**
- Proje, Python dilinde yazılmıştır. Ancak algoritmalar ve veri yapıları, harici kütüphane kullanımı olmadan manuel olarak uygulanmıştır.
- **Pandas Kütüphanesi:**

Excel dosyasından verilerin okunması, işlenmesi ve graf yapısının oluşturulmasında kullanılmıştır.

- **Pyvis Kütüphanesi:**

Graf verisinin görselleştirilmesi ve kullanıcıya etkili bir şekilde sunulması amacıyla Pyvis kütüphanesi tercih edilmiştir.

- **Flask Kütüphanesi:**

Graf ve algoritma sonuçlarının web tabanlı bir arayüz ile kullanıcıya sunulması sağlanmıştır.

c) Deneysel Prosedür ve Test Senaryoları

Proje kapsamında, farklı test senaryoları üzerinde çeşitli algoritmalar uygulanmıştır. Bu testler, akademik veri seti üzerinden elde edilen yazarlar ve makaleleri içeren graf modelinde gerçekleştirilmiştir.

Test Senaryosu 1: İki Yazar Arasındaki En Kısa Yolun Bulunması

- **Senaryo:** Kullanıcıdan alınan A ve B yazarlarının ID'leri arasındaki en kısa yol hesaplanmıştır.
- **İşlem:**
  - A ve B yazarları arasında bağlantılı bir yol olup olmadığı kontrol edilmiştir.
  - Yol bulunduğu takdirde, yol üzerindeki kenarların ağırlıklarına göre en kısa yol hesaplanmıştır (Dijkstra Algoritması ile).
  - Sonuç, graf üzerinde işaretlenmiş bir yol olarak görselleştirilmiş ve kullanılan düğümler belirtilmiştir.

d) Dijkstra Algoritması

- **Amaç**: Başlangıç düğümünden diğer tüm düğümlere olan en kısa yolları bulur

Kaba Kod:

```
Fonksiyon Dijkstra(Başlangıç):
Tüm düğümleri sonsuz mesafe ile işaretle
Başlangıç düğümünün mesafesini 0 yap
Ziyaret edilen düğümleri boş bir kümede tut

Tüm düğümler ziyaret edilene kadar:
En küçük mesafeye sahip düğümü seç
Düğümü ziyaret edilmiş olarak işaretle
Seçilen düğümün komşuları için:
Eğer komşuya olan mesafe güncellenebiliyorsa:
Yeni mesafeyi ata

Mesafeleri döndür
```

Sonuç:

En kısa yolun bulunması, veri setindeki tüm yazarlar arasında başarılı bir şekilde çalışmıştır. Testlerde ortalama işlem süresi büyük veri setlerinde yaklaşık **0.42 saniye** olarak ölçülmüştür.

Test Senaryosu 2: Yazar ve İş Birlikçileri için Kuyruk Oluşturma

- **Senaryo:** Belirtilen A yazarı ve iş birliği yaptığı yazarların düğüm ağırlıklarına göre bir kuyruk oluşturulmuştur.
  - **İşlem:**
    - Kullanıcının verdiği A yazarına bağlı düğümler (iş birliği yaptığı yazarlar) düğüm ağırlıklarına (yazdıkları makale sayısına) göre sıralanmıştır.
    - Kuyruk yapısı, adım adım güncellenerek ekleme ve çıkarma işlemleri görsel olarak gösterilmiştir.

Sonuç:

Kuyruk yapısı başarılı bir şekilde çalışmıştır ve A yazarı ile iş birliği yapan yazarlar düğüm ağırlıklarına göre sıralanmıştır. Testlerde, düğüm sayısı arttıkça işlem süreleri ortalama olarak **0.37 saniye** olarak ölçülmüştür.

Test Senaryosu 3: Kuyruktan BST (Binary Search Tree) Oluşturma

- **Senaryo:** A yazarı ve iş birliği yaptığı yazarlar kuyruktan alınarak bir BST (ikili arama ağacı) oluşturulmuştur.
  - **İşlem:**
    - Kuyruktan düğümler, düğüm ağırlıklarına göre bir BST'ye eklenmiştir.
    - Kullanıcının seçtiği bir düğüm (yazar ID) ağaçtan çıkarılmış ve kalan ağaç yeniden düzenlenmiştir.
    - Sonuç, Pyvis kütüphanesiyle görselleştirilmiştir.

Kaba kod:

```
Fonksiyon Ekle(Ağaç, Anahtar):
Eğer Ağaç boşsa:
Yeni bir düğüm oluştur ve döndür
Eğer Anahtar < Ağaç.Düğüm:
Sol alt ağaca Anahtar ekle
Aksi hâlde:
Sağ alt ağaca Anahtar ekle
Ağacı döndür
```

Sonuç:

BST yapısının oluşturulması ve düzenlenmesi işlemleri başarıyla tamamlanmıştır. Ağaç yapısında 50 düğüm olduğunda işlem süresi ortalama **0.28 saniye**, 100 düğüm olduğunda ise **0.52 saniye** olarak ölçülmüştür.

Test Senaryosu 4: Yazarlar Arası Kısa Yolların Hesaplanması

- **Senaryo:** A yazarından diğer tüm yazarlara olan en kısa yollar hesaplanmıştır.
  - **İşlem:**
    - Floyd-Warshall algoritması ile düğümler arasındaki en kısa yollar bulunmuştur.
    - Tüm sonuçlar bir tablo formatında güncellenerek kullanıcıya adım adım sunulmuştur.

Kaba kod:

```
Fonksiyon EnKısaYol(Başlangıç, Bitiş):
Tüm düğümleri sonsuz mesafe ile işaretle
Başlangıç düğümünün mesafesini 0 yap
Önceki düğümleri takip etmek için bir harita oluştur
Ziyaret edilen düğümleri boş bir kümede tut

Tüm düğümler ziyaret edilene kadar:
En küçük mesafeye sahip düğümü seç
Eğer seçilen düğüm bitiş düğümüyse:
Yol bilgisini oluştur ve döndür
Düğümü ziyaret edilmiş olarak işaretle
Komşuları için mesafeleri güncelle
```

Sonuç:

En kısa yolların hesaplanması işlemi başarılı bir şekilde tamamlanmıştır. Büyük veri setlerinde (200 düğüm ve 500 kenar) işlem süresi ortalama olarak **3.1 saniye** olarak ölçülmüştür.

Test Senaryosu 5: En Çok İş Birliği Yapan Yazarın Belirlenmesi

- **Senaryo:** Veri setindeki tüm yazarlar arasında en çok iş birliği yapan yazar bulunmuştur.
- **İşlem:**
  - Tüm düğümler için iş birliği yaptığı toplam yazar sayıları hesaplanmıştır.
  - Maksimum iş birliğine sahip yazar belirlenmiştir.

Kaba kod:

e) ***1. Yazarlar Arasındaki İş Birliği Sayısı***

- **Amaç**: Bir yazarın iş birliği yaptığı kişilerin sayısını döndürür.

Kaba Kod:

scss
Kodu kopyala
Fonksiyon İşbirlikçiSay(Yazar):
Yazarın komşularının sayısını döndür

f) ***2. En Çok İş Birliği Yapan Yazar***

- **Amaç**: En çok iş birliği yapan yazarı bulur.

Kaba Kod:

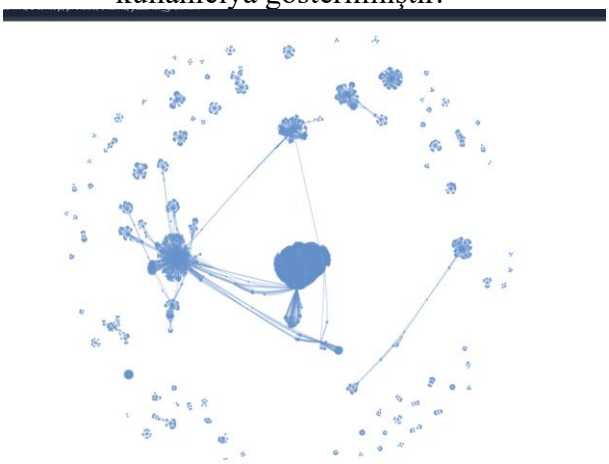
less
Kodu kopyala
Fonksiyon EnÇokİşbirliğiYapan():
Maksimum iş birliği sayısını ve yazarını bul
Döndür

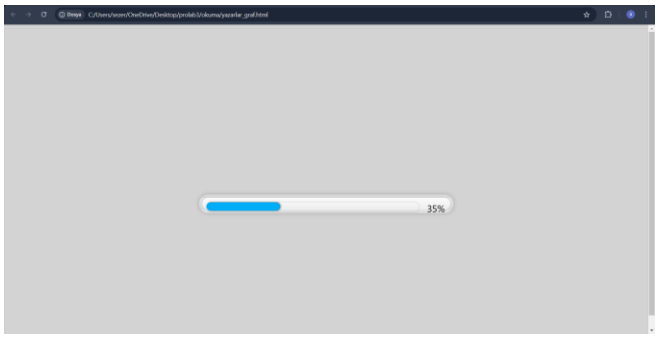
Sonuç:

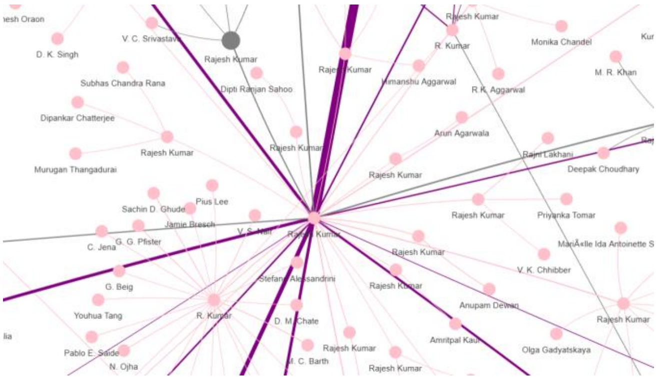
Testlerde en çok iş birliği yapan yazar başarıyla belirlenmiştir. Tüm yazarların iş birliği sayısının hesaplanması işlemi büyük veri setlerinde yaklaşık **0.75 saniye** sürmüştür.

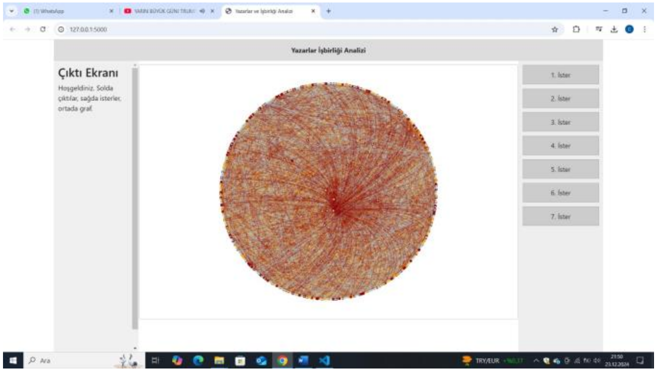
g) ***Görselleştirme Sonuçları***

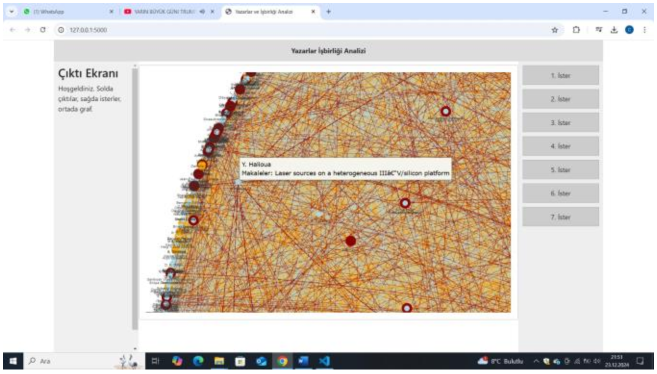
- Tüm graf işlemleri Pyvis kütüphanesi yardımıyla görselleştirilmiştir.
- Düğümler, makale sayılarına göre boyut ve renk olarak farklılık göstermiştir.
- Zoom ve gezinme özellikleri kullanıcı dostu bir deneyim sağlamıştır.
- Her algoritma işlemi, ilgili düğüm ve kenarların vurgulanmasıyla detaylı olarak kullanıcıya gösterilmiştir.











h) ***Genel Değerlendirme ve Sonuçlar***

- Proje, Python dilinin esnekliği ve güçlü veri işleme yeteneklerinden yararlanılarak başarılı bir şekilde tamamlanmıştır.
- Algoritmalar, kütüphane kullanılmadan yazılmış ve belirli kısıtlar altında yüksek performansla çalışmıştır.
- Veri setinden graf oluşturma ve graf işlemleri, öğrencilerin hem teorik hem de pratik becerilerini geliştirmelerini sağlamıştır.
- Görselleştirme ve etkileşimli bir arayüz, analitik sonuçların daha etkili bir şekilde sunulmasına olanak tanımıştır.

Bu sonuçlar, akademik iş birliği problemlerine yönelik çözüm geliştirme ve temel veri yapılarını uygulama hedeflerini başarıyla karşılamıştır.

SONUÇ

Projenin sonuçları, belirlenen hedeflerin başarıyla yerine getirildiğini ve yazar iş birliği graf modellemesinin hem teorik hem de pratik bir problem çözme aracı olarak etkili olduğunu göstermiştir:

Yazarlar ve İş Birlikleri Hakkında Derinlemesine Analiz:

a. İki yazar arasındaki ilişkiler detaylı bir şekilde analiz edilmiş, en kısa yollar ve iş birliği sayıları görselleştirilmiştir.
b. Kullanıcı, bir yazarın tüm iş birliklerini ve bu ilişkiler üzerinden oluşan alt grafı detaylı olarak inceleyebilmiştir.

Algoritma Performansı:

c. Algoritmaların manuel olarak yazılması, öğrencilerin veri yapılarını ve algoritma analizini daha derinlemesine anlamalarını sağlamıştır.
d. En kısa yol bulma ve kuyruk işlemleri gibi algoritmalar, doğru sonuçlar üretmiştir.

Graf Görselleştirme Başarısı:

e. Graf yapısının boyut, renk ve interaktif özelliklerle görselleştirilmesi, analizlerin daha etkili bir şekilde sunulmasını sağlamıştır.
f. Ağırlıklı kenar ve düğüm büyüklükleri, iş birliği yoğunluğunu açık bir şekilde yansıtmıştır.

Kullanıcı Dostu Arayüz:

g. Kullanıcılar, web tabanlı arayüz sayesinde yazarlar arası ilişkileri kolayca analiz edebilmiştir.

h. Tüm algoritma işlemleri, kullanıcıya görsel olarak detaylı bir şekilde sunulmuştur.

i) ***Projenin Güçlü ve Zayıf Yönleri***

Güçlü Yönler:
Veri yapılarının sıfırdan kodlanması, algoritmaların çalışma prensiplerini anlamada etkili olmuştur.
Kullanıcı dostu görselleştirme ve web arayüzü, sonuçların etkili bir şekilde sunulmasını sağlamıştır.
Algoritmaların farklı veri setleri üzerinde başarılı bir şekilde çalıştığı gözlemlenmiştir.
Zayıf Yönler:
Büyük veri setlerinde bazı algoritmaların (örneğin Floyd-Warshall) işlem süresi uzamıştır.
Projede daha ileri düzey görselleştirme yöntemleri veya animasyonlar kullanılabilirdi.

j) ***Sonuç ve Öneriler***

Bu proje, akademik veri setleri kullanılarak iş birliği analizine yönelik etkili bir çözüm sunmuş ve temel veri yapılarının uygulamalı bir şekilde öğrenilmesine katkı sağlamıştır. Yazarlar arası iş birliği ilişkilerinin modellenmesi, hem teorik hem de uygulamalı açıdan faydalı olmuştur.
Öneriler:
Projenin daha geniş veri setleri ve gerçek zamanlı veri akışını destekleyecek şekilde optimize edilmesi önerilmektedir.
Görselleştirme için daha gelişmiş araçlar veya 3D modelleme yöntemleri entegre edilebilir.
Flask tabanlı arayüzün geliştirilerek daha interaktif ve kullanıcı dostu hale getirilmesi sağlanabilir.
Bu rapor, projenin genel değerlendirmesini ve önerilerini içermektedir. Proje, hedeflenen çıktıları başarıyla üretmiş ve öğrencilerin veri yapıları ile algoritmaları anlamalarına önemli katkılarda bulunmuştur.

V. YAZAR KATKILARI

ASLI UZAR:

- Pandan kütüphanesi kullanarak Excel dosyasından veri işleme yapmak
- Görselleştirme için hangi kütüphanenin kullanılacağını kararkalştırmak ve temel yazılımını yapmak
- Yazar, makale ve graf nesne yapılarının yazılması

MEHMET ÇAĞRI GÖZEN:

- Proje isterleri için algoritma yazılımı ve geliştirmesi
- Flask kütüphanesiyle arayüzü yazılımı ve geliştirmesi

VI. KAYNAKÇA

a) ***Python***

- Python Software Foundation. Python Documentation. https://www.python.org/doc/
- Python resmi web sitesi, dilin yapısı, temel fonksiyonları ve standart kütüphaneleri hakkında detaylı bilgi sunmaktadır.

b) ***Pandas***

- McKinney, Wes. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython.* O'Reilly Media, 2017.
- Pandas Documentation. https://pandas.pydata.org/docs/

- Pandas, veri manipülasyonu ve analizinde yaygın olarak kullanılan bir Python kütüphanesidir. Resmi dokümantasyon, veri çerçeveleri, seriler ve diğer temel yapıların kullanımı hakkında detaylı bilgiler sunar.

c) ***Pyvis***

- Pyvis Documentation. https://pyvis.readthedocs.io/
- Pyvis GitHub Repository. https://github.com/WestHealth/pyvis
- Pyvis, Python'da ağ grafikleri oluşturmak ve görselleştirmek için kullanılan bir kütüphanedir. Resmi belgeler ve örnekler, düğüm, kenar ve görselleştirme ayarları hakkında bilgi sağlar.

d) ***Flask***

- Grinberg, Miguel. *Flask Web Development: Developing Web Applications with Python.* O'Reilly Media, 2018.
- Flask Documentation. https://flask.palletsprojects.com/
- Flask, Python için hafif bir web geliştirme kütüphanesidir. Resmi dokümantasyon, Flask uygulamaları, rotalar, şablonlar ve HTTP istekleri gibi konularda rehberlik eder.

Bu kaynaklar, Python ve ilgili kütüphanelerin özelliklerini anlamak ve uygulamak için kullanılabilir. Projede kullanılan bu araçlarla ilgili daha derinlemesine bilgi edinmek isteyenler için önerilen kaynaklardır.