ÜNİVERSİTE KAMPÜSÜ İÇERİSİNDEKİ FAKÜLTE ROUTERLARI ARASINDAKİ EN KISA MESAFENİN BELİRLENMESİ

FAİK DÖNER 210542001
EMİR YILDIRIM 210541069

1. PROJE ÖZETİ

Bu projede, üniversite kampüsü içinde yer alan farklı fakültelerin routerları arasındaki en kısa mesafeyi belirlemeyi hedefliyoruz. Üniversite kampüsü, geniş bir alana yayılan çok sayıda fakülte ve bu fakülteleri birbirine bağlayan ağ altyapısından oluşmaktadır. Bu altyapının etkin ve verimli bir şekilde yönetilebilmesi için routerlar arasındaki en kısa yolların belirlenmesi kritik önem taşımaktadır.
Projede, Dijkstra ve Bellman-Ford algoritmalarını kullanarak routerlar arasındaki en kısa mesafe hesaplamalarını gerçekleştireceğiz.

2. PROJE AMACI

Bu proje de, üniversite kampüsünde bulunan farklı fakültelerin routerları arasında en kısa mesafeyi belirleyerek, ağ trafiğini optimize etmek ve veri iletim hızını artırmaktır. Üniversite kampüsleri, geniş alanlara yayılan ve farklı binalar arasında karmaşık ağ bağlantıları gerektiren yerleşkelerdir. Bu ağ bağlantılarının etkin bir şekilde yönetilmesi, ağ performansının iyileştirilmesi ve veri iletim sürelerinin minimize edilmesi açısından kritik öneme sahiptir.
Bu amaç doğrultusunda, hem Dijkstra hem de Bellman-Ford algoritmalarını kullanarak en kısa yol hesaplamalarını gerçekleştireceğiz. Bu iki algoritmayı kullanarak, kampüs ağ yapısındaki farklı senaryoları değerlendirme ve en uygun çözümü belirleme fırsatına sahip olacağız.

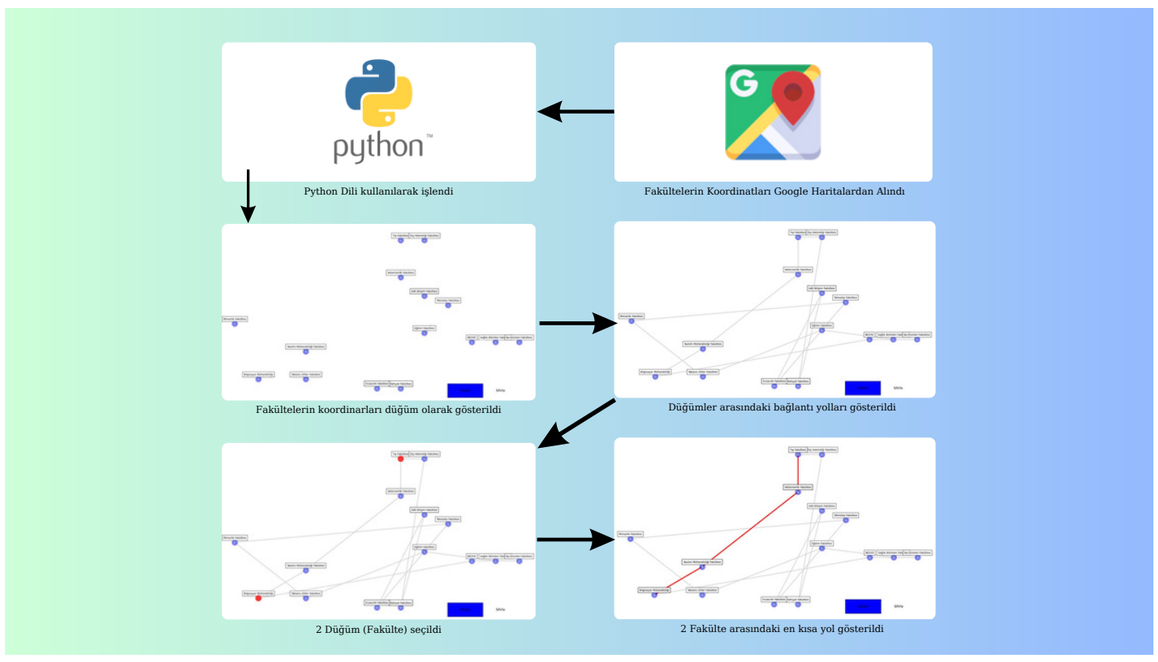
3. PROJE KAPSAMI

1- Veri Toplama ve Graf Oluşturma:
Üniversite kampüsündeki tüm fakültelerin ve bu fakültelere ait routerların konumlarını ve bağlantı bilgilerini toplayarak routerlar arasında bir yönlü ağırlıklı graf oluşturacağız.
2- Algoritmaların Uygulanması:
Toplanan veriler kullanılarak oluşturulan graf üzerinde Dijkstra ve Bellman-Ford algoritmalarını ayrı ayrı uygulayacağız.
Her iki algoritmanın da başlangıç noktası olarak belirlenen bir routerdan diğer tüm routerlara olan en kısa mesafeleri hesaplanacaktır.
3- Performans Karşılaştırması:
Hesaplamalar tamamlandıktan sonra, her iki algoritmanın performansını çeşitli kriterler (örneğin, hesaplama süresi, bellek kullanımı) açısından karşılaştıracaktır.
4- Sonuçların Değerlendirilmesi ve Yorumlanması:
Hesaplanan en kısa mesafeler ve algoritmaların performans sonuçları üzerinden bir değerlendirme yapacağız.
Üniversite kampüsünün ağ altyapısının optimizasyonu ve verimliliği açısından hangi algoritmanın daha uygun olduğunu belirlenecektir.

4. YÖNTEMLER

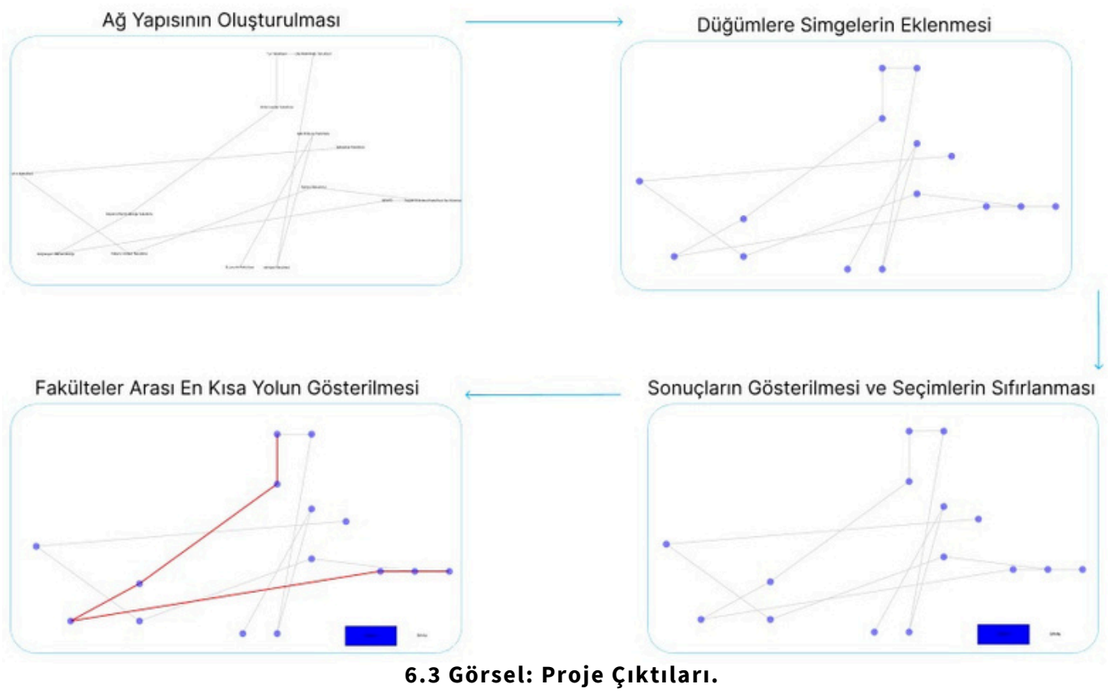
1- Kullanlan Yöntemler:
Ağın Oluşturulması: NetworkX kütüphanesi kullanılarak boş bir ağ oluşturulmuş ve düğümler (fakülteler) bu ağa eklenmiştir. Düğümler arası rastgele bağlantılar pozitif ağırlıklı kenarlarla eklenmiştir.
Manuel Düğüm Pozisyonları: Düğümlerin pozisyonları manuel olarak belirlenmiş ve grafik üzerinde doğru yerlerde gösterilmiştir.
Bilgisayar Simgesi Ekleme: Grafik üzerine bilgisayar simgesi eklemek için Matplotlib'in OffsetImage ve AnnotationBbox sınıfları kullanılmıştır. Her bir düğümün üzerine fakülte adları eklenmiştir.
Ağın Görselleştirilmesi: NetworkX ve Matplotlib kullanılarak ağın düğümleri ve kenarları görselleştirilmiş, düğümler üzerindeki fakülte adları gösterilmiştir. Düğümler tıklanabilir hale getirilmiştir.
Animasyon: Matplotlib'in FuncAnimation sınıfı kullanılarak düğümlerin animasyonlu gösterimi sağlanmıştır. Her 0.65 saniyede bir düğüm güncellenmiştir.
Fakülteler Arası En Kısa Yol: Kullanıcı tarafından seçilen iki fakülte arasındaki en kısa yol, Dijkstra algoritması kullanılarak hesaplanmış ve kırmızı renkle vurgulanmıştır.
Kullanıcı Etkileşimi: Matplotlib.widgets kullanılarak "Göster" ve "Sıfırla" butonları eklenmiş ve bu butonlara tıklama olayları bağlanmıştır. Kullanıcı tıklamaları dinlenerek seçilen fakülteler arasında en kısa yolun gösterilmesi sağlanmıştır.
2- Kullanılan Kütüphaneler:
networkx: Bu kütüphane, grafik ve ağ yapılarıyla çalışmak için kullanılır. Düğümler ve kenarlarla temsil edilen grafiklerin oluşturulması, yönetilmesi ve analiz edilmesi için kullanılır.
matplotlib.pyplot: Bu kütüphane, grafiklerin ve görselleştirmelerin oluşturulması için kullanılır. Grafikteki düğümler, kenarlar ve diğer grafik elemanları bu kütüphane ile çizilir.
matplotlib.offsetbox: Grafik üzerine görüntü ve metin gibi öğeleri eklemek için kullanılır. Projede bilgisayar simgesi ve fakülte adlarını eklemek için kullanılmıştır.
matplotlib.widgets: Grafik arayüzü için buton gibi etkileşimli öğeler oluşturmak için kullanılır.
matplotlib.animation: Animasyonlu grafikler oluşturmak için kullanılır. Projede ağın düğümlerinin animasyonlu olarak gösterilmesi için kullanılmıştır.
numpy: Sayısal hesaplamalar ve rastgele sayı üretimi gibi işlemler için kullanılır.

6. GÖRSEL VE GRAFİKLER



6.1 Görsel: Graphical Abstract.

6.2 Görsel: Akış Diyagramı.



6.3 Görsel: Proje Çıktıları.

5. SONUÇ VE ÖNERİ

Dijkstra algoritmasının pozitif ağırlıklı graflarda daha hızlı ve verimli olduğunu, Bellman-Ford algoritmasının ise negatif ağırlıklı graflarda kullanılabileceğini göstermektedir. Gelecekte, daha büyük ağ yapıları ve farklı algoritmalar üzerinde çalışılması önerilmektedir.
Dijkstra algoritması daha hızlı ve verimli çalışırken, Bellman-Ford algoritması negatif ağırlıkları ve döngüleri yönetebilme kabiliyeti ile öne çıkar.
Her iki algoritmanın da performansını karşılaştırarak, hangi durumlarda kullanılabileceğini belirleyebiliriz.

7. KAYNAKÇA

- https://www.google.com/search//firat+üniversitesi+fakülteler+haritası
- https://logos-world.net/python-logo/
- Dijkstra, E. W. (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*
- Bellman, R. (1958). On a Routing Problem. *Quarterly of Applied Mathematics*, 16, 87-90.

BİZE ULAŞINIZ

Faik DÖNER
Yazılım Mühendisliği 3. sınıf
Tel. No: 0530 299 9564
E-mail: faikardad@gmail.com
Github:
github.com/Faikdnr/RouterGoruntuleme

Emir YILDIRIM
Yazılım Mühendisliği 3.sınıf
Tel. No: 0544 494 8651
E-mail: emir76931@gmail.com
Github:
github.com/emir2323/RouterGoruntuleme