

## 1. Şehirler Arası Rota Planlama ve Optimizasyon Sistemi

Bu projede, şehirler arası ulaşım için akıllı bir rota planlama sistemi geliştirilecektir. Sistem, farklı rota bulma algoritmalarını destekleyecek ve görsel bir arayüz sunacaktır.

### 1. Sınıf Yapısı ve Gereksinimler

- **RotaStratejisi:** Soyut temel sınıf (Abstract Base Class) olarak tanımlanmalıdır
- **Konum:** Şehirleri ve bağlantılarını temsil eden sınıf
  - Komşu şehirler ve mesafeler **komsular** sözlüğünde tutulmalıdır
  - Format: {Konum: mesafe} şeklinde olmalıdır
- **Rota Algoritmaları:**
  - Dijkstra: En kısa mesafe hesaplaması için
  - BFS: En az durak sayısı hesaplaması için
- **Harita:** Tüm sistemi yöneten ana sınıf
  - Şehir ve bağlantı yönetiminden sorumludur
  - Varsayılan olarak çift yönlü bağlantıları desteklemelidir

### 2. Sistem Özellikleri

- Dinamik şehir ve bağlantı ekleme özelliği
- Çoklu rota hesaplama stratejisi desteği
- Dijkstra algoritması ile toplam mesafe optimizasyonu
- BFS algoritması ile minimum durak sayısı hesaplaması
- Genişletilebilir strateji mimarisi
- Matplotlib ile interaktif görselleştirme
- `set_strateji()` metodu ile algoritma değişimi

### 3. Görselleştirme Gereksinimleri

- Başlık ve eksenler (`plt.title`, `plt.axis`)
- Çizim alanı boyutu 12x8 inç (`plt.figure`)
- Şehirler arası bağlantılar: Gri kesikli çizgiler (`plt.plot`)
- Şehir konumları: Kırmızı noktalar (`plt.scatter`)
- Görsel elemanlar:
  - Şehir isimleri
  - Mesafe bilgileri
  - Açıklayıcı başlık

- Gizlenmiş eksen çizgileri
- Maliyet değerleri bağlantı çizgilerinin ortasında

#### 4. Örnek Çıktılar

- Örnek kullanım:

```
if __name__ == "__main__":
    harita = Harita()
    harita.konum_ekle(Konum("İstanbul", x=-20, y=60))
    harita.konum_ekle(Konum("Ankara", x=50, y=50))
    harita.konum_ekle(Konum("İzmir", x=-30, y=-30))
    harita.konum_ekle(Konum("Antalya", x=20, y=-40))
    harita.konum_ekle(Konum("Bursa", x=10, y=20))
    harita.konum_ekle(Konum("Konya", x=30, y=0))
    harita.konum_ekle(Konum("Adana", x=80, y=-20))
    harita.konum_ekle(Konum("Trabzon", x=100, y=70))

    # Bağlantıları ve maliyetleri ekle
    harita.baglanti_ekle("İstanbul", "Ankara", 450)
    harita.baglanti_ekle("İstanbul", "Bursa", 150)
    harita.baglanti_ekle("İstanbul", "İzmir", 480)
    harita.baglanti_ekle("Ankara", "Konya", 260)
    harita.baglanti_ekle("Ankara", "Trabzon", 780)
    harita.baglanti_ekle("İzmir", "Antalya", 420)
    harita.baglanti_ekle("Antalya", "Konya", 320)
    harita.baglanti_ekle("Konya", "Adana", 340)
    harita.baglanti_ekle("Bursa", "Ankara", 380)
    harita.baglanti_ekle("Bursa", "İzmir", 330)
    harita.baglanti_ekle("Adana", "Trabzon", 890)

    # Dijkstra testi (En düşük maliyet)
    harita.set_strateji(Dijkstra())
    yol, maliyet = harita.rota_bul("İstanbul", "Trabzon")
    print(f"Dijkstra ile en kısa yol: {yol}, Toplam maliyet: {maliyet}")

    # BFS testi (En az aktarma)
    harita.set_strateji(BFS())
    yol, aktarma = harita.rota_bul("İstanbul", "Trabzon")
```

```
print(f"BFS ile en az aktarmalı yol: {yol},  
Aktarma sayısı: {aktarma}")
```

```
# Görselleştirme  
harita.visualize()
```

- Program çıktısı:

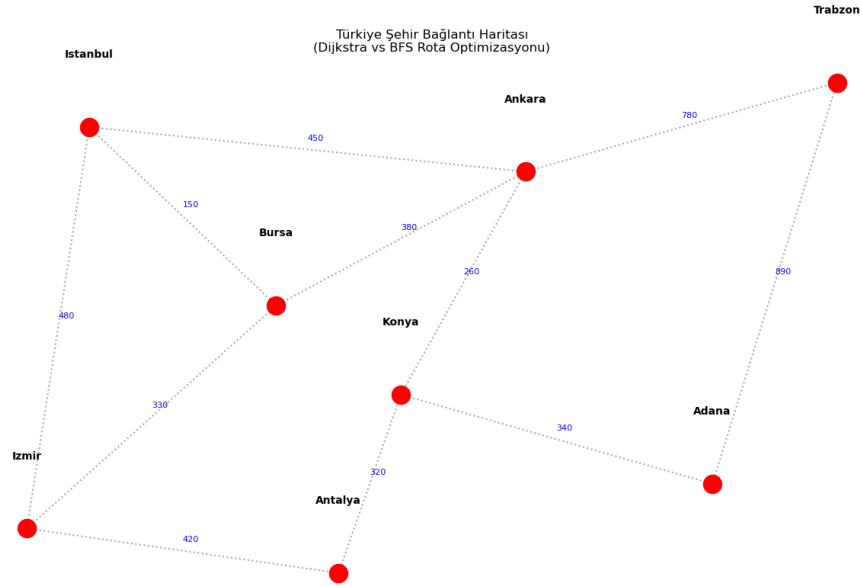
Dijkstra ile en kısa yol: ['Istanbul', 'Ankara', 'Trabzon'],

Toplam maliyet: 1230

BFS ile en az aktarmalı yol: ['Istanbul', 'Ankara', 'Trabzon'],

Aktarma sayısı: 2

- Görselleştirme



2. Projenizi GitHub'a yükleyip, bağlantısını k.yavuzkurt1@gmail.com adresine gönderiniz.