



Global AI Hub

Akbank Python ile Yapay Zekaya Giriş Bootcamp

Proje Dosyası

Sürücüsüz Metro Simülasyonu (Rota Optimizasyonu)

Bu projede, bir metro ağında;

- * İki istasyon arasındaki en hızlı,
- * En az aktarmalı rotayı bulan bir simülasyon geliştireceksiniz.

Projenin isterlerini yerine getirmek kaydıyla, daha büyük bir graph kullanmak, görselleştirme eklemek, farklı özellikler eklemek vb. Yollarla projenizi zenginleştirebilirsiniz.



Proje Amacı

Bu projede aşağıdaki hedeflere ulaşmanız beklenmektedir:

1. Graf veri yapısını kullanarak metro ağını modelleme
2. BFS (Breadth-First Search) algoritması ile en az aktarmalı rotayı bulma
3. A* algoritması ile en hızlı rotayı bulma
4. Gerçek dünya problemlerini algoritmik düşünce ile çözme



Görev Listesi

1. Projeyi İnceleme ve Anlama

- metro_simulation.py dosyasını açın ve kodun yapısını inceleyin. Dosya ismini “AdSoyad_MetroSimulation.py” şeklinde güncelleyin. **Uygun formatta olmayan dosya isimleri elenmenize sebep olur.**
- İstasyon ve MetroAgi sınıflarının yapısını ve metodlarını anlayın
- Test senaryolarını ve örnek metro ağı yapısını inceleyin

2. Algoritmaları Tamamlama

BFS Algoritması (en_az_aktarma_bul)

1. Fonksiyonun amacını ve parametrelerini anlayın
2. BFS algoritmasının nasıl çalıştığını araştırın
3. Verilen ipuçlarını kullanarak algoritmayı tamamlayın:
 - * Kuyruk yapısını oluşturun (collections.deque)
 - * Ziyaret edilen istasyonları takip edin
 - * Komşu istasyonları keşfedin
 - * En kısa rotayı bulun

A* Algoritması (en_hizli_rota_bul)

1. Fonksiyonun amacını ve parametrelerini anlayın
2. A* algoritmasının nasıl çalıştığını araştırın
3. Verilen ipuçlarını kullanarak algoritmayı tamamlayın:
 - * Öncelik kuyruğunu oluşturun (heapq)
 - * Ziyaret edilen istasyonları takip edin
 - * Toplam süreyi hesaplayın ve en hızlı rotayı bulun

3. Test ve Doğrulama

- Kodunuzu çalıştırın ve test senaryolarının sonuçlarını kontrol edin
- Farklı başlangıç ve bitiş noktaları için testler yapın
- Bulunan rotaların mantıklı olup olmadığını kontrol edin

4. Dokümantasyon

Projenizi kendi GitHub reponuzda paylaşırken şu bilgileri içeren bir README.md dosyası hazırlayın:

1. Proje Başlığı ve Kısa Açıklama
2. Kullanılan Teknolojiler ve Kütüphaneler (önceden import edilmiş ve kullanılmış heapq, collections gibi kütüphaneleri de açıklamanızı bekliyoruz)
3. Algoritmaların Çalışma Mantığı
 - BFS algoritmasının nasıl çalıştığı
 - A* algoritmasının nasıl çalıştığı
 - Neden bu algoritmaları kullandığımız
4. Örnek Kullanım ve Test Sonuçları
5. Projeyi Geliştirme Fikirleri

5. GitHub'da Paylaşma

1. Yeni bir GitHub repository oluşturun
2. Repository'yi "Public" olarak ayarlayın
3. Kodunuzu ve README.md dosyanızı yükleyin
4. Commit mesajlarınızı açıklayıcı yazın



Değerlendirme Kriterleri

Projeniz aşağıdaki kriterlere göre değerlendirilecektir:

1. **Algoritma Implementasyonu**
 - BFS algoritmasının doğru çalışması
 - A* algoritmasının doğru çalışması

2. Kod Kalitesi

- Kodun okunabilirliği ve düzeni
- Değişken ve fonksiyon isimlendirmeleri
- Hata kontrolü ve güvenlik

3. Dokümantasyon

- README.md dosyasının detaylı ve açıklayıcı olması
- Kodun içindeki yorum satırları
- Algoritmaların çalışma mantığının açıklanması



Faydalı Kaynaklar

1. BFS Algoritması:

- [Python BFS Implementation](#)
- [BFS Algorithm Visualization](#)

2. A* Algoritması:

- [A* Pathfinding Algorithm](#)
- [A* Search Algorithm in Python](#)

3. Kütüphaneler

- [Python Collections Module](#)
- [Python Heapq Module](#)



Yardım ve İletişim

Eğer takıldığınız bir nokta olursa:

1. Önce bu dokümantasyonu tekrardan dikkatlice okuyun.
2. Fikir alışverişi için DeepSeek, ChatGPT gibi araçları kullanmanıza pek tabii ki engel olamayız. Öğrenme yolculuğunuzun sağlığı ve sertifikasyon süreciniz için, copy + paste yapmaktan kaçının.
3. Size özel kanallarınızda, size yardımcı olmak için hazır bulunan mentörlerinize danışın.

İyi çalışmalar! 🚀