

Proje Başlığı: Online Kargo Takip Sistemi

Amaç: Öğrencilerin farklı veri yapıları ve algoritmaları entegre ederek gerçek bir problem üzerinde çalışmaları.

Proje Özellikleri ve Teknik Detaylar

1. Müşteri Verilerinin Yönetimi (Linked List)

Her müşteri için aşağıdaki bilgiler tutulmalıdır:

- **Müşteri ID** (benzersiz bir kimlik numarası)
- **İsim ve Soyisim**
- **Kargo Gönderim Geçmişi** (Linked List kullanılarak)
 - Her bir gönderi aşağıdaki bilgilere sahip olmalıdır:
 - Gönderi ID
 - Gönderi Tarihi
 - Teslim Durumu (Teslim Edildi / Teslim Edilmedi)
 - Teslim Süresi (gün cinsinden)

Görevler:

- Yeni bir müşteri ekleme.
- Mevcut bir müşterinin gönderim geçmişini sorgulama.
- Gönderi geçmişi sonradan eklendiği için **düğümüleri sıralı bir şekilde ekleme** işlemi yapılmalı (örneğin tarih sırasına göre).

2. Kargo Önceliklendirme (Priority Queue)

Sistem, kargoları teslimat önceliğine göre işleme almalıdır.

- **Gerekli Bilgiler:**
 - Gönderi ID
 - Teslimat Süresi
 - Kargo Durumu (İşleme Alındı / Teslimatta / Teslim Edildi)

Priority Queue Uygulaması:

- Teslimat süresi daha kısa olan kargolar öncelikli olmalıdır.
- Kargo ekleme ve öncelikli kargo işleme işlemleri için zaman karmaşıklığı analiz edilmelidir.

3. Kargo Rotalama (Tree Data Structure)

Teslimat yapılacak şehirler ve rotalar bir ağaç yapısı ile temsil edilecektir.

- **Ağaç Yapısı:**
 - Kök düğüm: Kargo şirketinin merkezi.
 - Alt düğümler: Şehirler ve alt rotalar.

Özellikler:

- Her düğüm bir şehir olacak ve aşağıdaki bilgilere sahip olacaktır:
 - Şehir Adı
 - Şehir ID
 - Şehre bağlı alt şehirler (çocuk düğümler)
 - Sistem, ağacın **derinliğini** kullanarak en kısa teslimat süresini hesaplamalıdır.
 - Rotaların görselleştirilmesi için öğrencilerden **ağacın bir çıktısını** üretmeleri istenebilir (örneğin, konsol üzerinde bir ağaç yapısı).
-

4. Gönderim Geçmişi Sorgulama (Stack)

Her müşteri için son gönderilen kargoları hızlıca sorgulamak için bir **yığın (stack)** kullanılmalıdır.

İşlevler:

- Son gönderilen 5 kargoyu sorgulama.
 - Yeni gönderim eklenmesi durumunda **stack'e push işlemi** yapılmalıdır.
 - Eğer gönderim geçmişi boşsa, uygun bir hata mesajı döndürülmelidir.
-

5. Kargo Durum Sorgulama (Sorting & Searching)

Sistem, kargo durumlarını sorgulamak için sıralama ve arama algoritmalarını kullanmalıdır.

- **Teslim Edilmiş Kargolar:**
 - Kargo ID'ye göre **binary search** algoritması kullanılarak bulunmalıdır.
 - Arama işlemi sırasında **sorted list** kullanılmalıdır.
- **Teslim Edilmemiş Kargolar:**

- Teslimat süresine göre **merge sort** veya **quick sort** kullanılarak sıralanmalıdır.
- Sıralama işleminin zaman karmaşıklığı analiz edilmelidir.

Ek Özellikler

6. Raporlama ve Performans Analizi

Öğrencilerden, kullandıkları veri yapıları ve algoritmaların performansını raporlamaları istenmelidir:

- Hangi veri yapısı neden seçildi?
- Algoritmaların zaman ve uzay karmaşıklığı.
- Daha verimli bir çözüm önerilebilir mi?

7. Arayüz (Opsiyonel)

Projeyi daha görselleştirmek için bir **konsol tabanlı menü** veya basit bir GUI oluşturulabilir:

- Menü Seçenekleri:
 1. Yeni müşteri ekle.
 2. Kargo gönderimi ekle.
 3. Kargo durumu sorgula.
 4. Gönderim geçmişini görüntüle.
 5. Tüm kargoları listele (sıralı).
 6. Teslimat rotalarını göster.

Teslimat Gereksinimleri

1. Kod:

- Projenin eksiksiz ve çalışan bir kodu teslim edilmelidir.
- Kod, her veri yapısını ve algoritmayı belirgin bir şekilde içermelidir.

2. Dokümantasyon:

- Algoritmaların ve veri yapılarının seçimi.
- Performans analizi ve sonuçları.
- Kullanıcı rehberi.

3. Test Senaryoları:

- Çeşitli senaryolarla sistem test edilmelidir (örneğin, kargo ekleme, arama, sıralama).

Zorluk Düzeyi ve Değerlendirme Kriterleri

- Proje zorluğu, farklı veri yapılarının bir arada kullanımı ve entegrasyonu ile artırılmıştır.
- Değerlendirme:
 1. Kodun doğruluğu (%40)
 2. Algoritmaların ve veri yapıların doğru kullanımı (%30)
 3. Performans analizi ve dokümantasyon (%30)