Grup Uyeleri:

1. Yağızhan Can - 21120205711

2. Furkan Topçu - 24120205080

3. aleyna ölmez - 24120205079

Online Kargo Takip Sistemi

Bu proje, PySimpleGUI kullanarak geliştirilmiş, farklı veri yapılarını (Linked List, Priority Queue, Tree, Stack) ve çeşitli algoritmaları (Sıralama, Binary Search, BFS vb.) entegre eden bir Online Kargo Takip Sistemi örneğidir. Kod, müşteri ve kargo yönetiminden, ağaç yapısında rota hesabına kadar bir dizi işlevi gerçekleştirerek gerçek bir senaryoyu modellemeyi amaçlar.

Proje Hakkında

Bu proje, bir kargo firmasının online kargo takip sistemini temel alarak:

Müşteri verilerinin linked list ile yönetilmesini,

Kargoların önceliklendirilmesini (Priority Queue - Min Heap),

Teslimat rotalarının ağaç yapısında modellenmesini,

Çeşitli sorgulama ve sıralama işlemlerinin uygulanmasını,

PySimpleGUI tabanlı bir arayüz (GUI) ile kullanıcı etkileşimini,

Ve son olarak, kullanıcı işlemlerinin (transaction\_history) kaydını sunar.

Projenin temel amacı; akademik ve pratik açıdan farklı veri yapılarını ve algoritmaları bir arada uygulayarak bir kargo takip senaryosu üzerinden deneyim kazanılmasını sağlamaktır.

Özellikler

1. Müşteri Yönetimi (Linked List)

Müşteriler, bağlı liste (linked list) yapısı kullanılarak saklanır.

Her müşteri, kendisine ait Gönderi Geçmişi (yine bir linked list) ve Son 5 Gönderi (stack mantığı) bilgilerini tutar.

Eklenen Yenilik: Müşterilerin eklendiği/ silindiği her adım, transaction\_history listesine kaydedilir.

Neden Linked List?

Dinamik veri ekleme/silme gerektiren senaryolarda linked list, dizilere (array) göre daha esnektir.

Özellikle gönderi geçmişi tarihe göre sıralı eklendiğinde, dizilerdeki kaydırma maliyeti yerine, linked list’te düğüm bazlı ekleme yapılabilir.

2. Gönderi Geçmişi ve Sıralı Ekleme

Müşteri başına bir ShipmentLinkedList bulunur. Gönderiler tarih sırasına göre eklenir.

Son 5 gönderi bilgisi, stack (liste) yapısıyla O(1) amortize ekleme/silme imkânı sağlar. Bu sayede “bir müşterinin son 5 gönderisini göster” gibi sorgulamalar kolaylaşır.

3. Kargo Önceliklendirme (Priority Queue - Min Heap)

Kargoların teslim süresine göre öncelikli ele alınması için heapq modülüyle bir min-heap (priority queue) kullanılmıştır.

Neden Min Heap?

En kısa teslim süresine sahip kargoyu en hızlı şekilde işleme alabilmek için idealdir.

Ekleme ve silme (pop) işlemleri ortalama O(log n) karmaşıklığa sahiptir.

4. Kargo Rotalama (Tree Yapısı)

Teslimat yapılacak şehirler ve rotalar, CityNode adlı bir ağaç yapısıyla modellenir.

Kök düğüm, kargo şirketinin merkezini temsil eder, alt düğümler farklı şehirlere giden yolları.

BFS ile en kısa rota derinliği (shortest\_route\_depth) hesaplanarak rota süreleri hakkında fikir edinilebilir.

Neden Ağaç?

Ağaç, şehir ve alt-şehir gibi hiyerarşik yapıları doğal şekilde temsil eder.

BFS kullanarak O(n) sürede en kısa rota derinliğini bulmak mümkündür.

5. Gönderim Geçmişi Sorgulama (Stack)

Her müşteri için, son 5 gönderi stack (LIFO) mantığıyla saklanır.

Yeni bir gönderi geldiğinde en üste (push) eklenir; 5’ten fazla gönderi varsa en alt (en eski) çıkarılır.

Bu yapı, son gönderilere hızlı erişim sağlayarak 7 numaralı menü seçeneğiyle anında gösterilebilmesini mümkün kılar.

6. Kargo Durum Sorgulama (Sıralama & Arama)

Teslim Edilmiş Kargolar: ID’ye göre sorted list oluşturulur ve binary search (ikili arama) ile O(log n) sürede aranır.

Teslim Edilmemiş Kargolar: Teslim süresine göre merge sort ile sıralanır (O(n log n)).

7. İşlem Geçmişi (Transaction History)

Uygulamadaki her önemli işlem, transaction\_history adlı bir listede saklanır.

Menüde 21 numaralı seçenek ile “İşlem Geçmişi” penceresi açılarak, yapılan tüm işlemler (müşteri ekleme, kargo sorgulama, vb.) sırasıyla görüntülenebilir.

8. Zaman Karmaşıklığı ve ASCII Grafikleri

Kod içerisinde her menü seçeneğinin karmaşıklığı print\_complexity fonksiyonuyla popup olarak gösterilir.

ASCII grafiği (örneğin \*\*\*\*) ile sembolik bir karmaşıklık düzeyi yansıtılır.

9. Basit Ama Gelişmiş Arayüz (PySimpleGUI)

Komut satırı yerine PySimpleGUI üzerinden buton/tabanlı bir arayüz sunar:

Menüler (Müşteri işlemleri, Kargo işlemleri vb.),

Popup pencereler (ör. kargo ekleme formu),

Scrollable text box (ör. işlem geçmişini veya listelediğimiz kargoları göstermek için)

Menü Seçenekleri:

Yeni müşteri ekle

Kargo gönderimi ekle

Kargo durumu sorgula

Gönderim geçmişini görüntüle

Tüm kargoları listele

Teslimat rotalarını göster

Bir müşterinin son 5 gönderisini göster

Öncelikli kargoları görüntüle (PQ)

Müşteri sil

Kargoyu Teslim Et

Ağaç yapısına yeni şehir ekle

En uzun teslim süreli teslim edilmiş kargo

Müşterileri isim sırasına göre listele

Müşteri ismine göre ara

Teslim edilmemiş kargo sayısı

Toplam kargo sayısı

Teslim edilmiş kargoların ortalama teslim süresi

Toplam müşteri sayısı

Tüm teslim edilmemiş kargoları sil

Çıkış

İşlem Geçmişi (yeni eklenen menü ile yapılan tüm işlemleri görme)

Performans Analizi

Linked List: Müşteri veya gönderi ekleme/silme/arama işlemleri ortalama O(n).

Priority Queue (heap): Kargo ekleme/silme (pop) işlemleri ortalama O(log n).

Binary Search (teslim edilmiş kargolarda ID arama): O(log n).

Merge Sort (teslim edilmemiş kargoların teslim süresine göre sıralanması): O(n log n).

Ağaç (Tree): BFS ile O(n)’de en kısa rota derinliği.

Stack: Son 5 gönderi ekleme/görüntüleme O(1) amortize.

Nasıl Kullanılır?

Kurulum

bash'in icine asagidaki kod yazilarak

pip install PySimpleGUI

Ardından, Python 3 ile bu dosyayı çalıştırabilirsiniz.

Çalıştırma

bash'in icine asagidaki kod yazilarak

python <dosya\_adi.py>

Uygulama, PySimpleGUI menü penceresiyle açılır.

Menü Seçenekleri

İlgili menü öğesine tıklayarak müşteri ekleme, kargo ekleme, kargo sorgulama gibi işlemlerinizi yapabilirsiniz.

Her işlemden sonra, transaction\_history güncellenir ve menüde 21 numarayla açılabilen “İşlem Geçmişi” penceresinde gözlemlenebilir.

Örnek Senaryolar

Müşteri Ekleme

Menüde “1. Yeni müşteri ekle” seçerek, ID, İsim, Soyisim girilmesiyle yeni bir müşteri oluşturulur.

Kargo Ekleme

“2. Kargo gönderimi ekle” menüsünden, Müşteri ID’siyle ilişkili olarak bir kargo ekleyebilirsiniz. Teslim süresi, durumu, tarihi vb. bilgileri girerek ekleme yaparsınız.

Eğer teslim durumu “Teslim Edilmedi” ise undelivered\_shipments listesine ve priority queue’ya eklenir.

“Teslim Edildi” ise delivered\_shipments listesine atılır.

Kargo Durumu Sorgulama

“3. Kargo durumu sorgula” ile, teslim edilmiş kargo ID’sini girerek binary search üzerinden arama yapabilirsiniz.

Gönderim Geçmişi

“4. Gönderim geçmişini görüntüle” menüsünden, belirli bir müşterinin tüm gönderim geçmişini;

“7. Bir müşterinin son 5 gönderisini göster” menüsünden, stack’te saklanan son 5 gönderiyi görebilirsiniz.

Kargo Listeleme

“5. Tüm kargoları listele” ile, teslim edilmiş ve edilmemiş kargolar sırasıyla ekrana basılır (ID veya teslim süresine göre).

Rota Gösterimi

“6. Teslimat rotalarını göster” menüsünde ağaç yapısı (şehirler) hiyerarşik olarak listelenir, BFS ile en kısa rota derinliği hesaplanır.

İşlem Geçmişi

“21. İşlem Geçmişi” sekmesinde, uygulama boyunca yapılan tüm işlemlerin kaydı görüntülenebilir.

Sonuç

Bu Online Kargo Takip Sistemi:

Farklı veri yapılarının (Linked List, Heap, Tree, Stack) gerçek bir problem üzerinden kullanımını gösterir.

PySimpleGUI ile kullanıcı dostu bir arayüz sağlar.

Zaman karmaşıklığı analizini popup hâlinde vererek eğitsel bir deneyim sunar.

transaction\_history listesiyle yapılan her işlemi kaydeder ve sonradan incelenebilmesine imkân tanır