

Kocaeli Üniversitesi Bilgisayar Mühendisliği

Bölümü Programlama Laboratuvarı I - Proje III

İbrahim İnce
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü
2. Öğretim
200202102

I. ÖZET

Öncelikli kuyruk veri yapılarının kullanıldığı bu projede, iniş ve kalkış pistleri olmak üzere iki adet uçak pisti olan bir havalimanının uçuş yönetim sistemi hazırlanmıştır.

Pistlerdeki uygun boş saatlere göre iniş izni isteyen uçakların izinlerinin onaylanması veya reddedilmesi, iniş izni talep eden uçakların öncelikleri göz önünde bulundurularak yapılmıştır.

II. GİRİŞ

Proje C programlama dili ve Code::Blocks IDE'si kullanılarak hazırlanmıştır.

Havalimanına iniş yapacak uçakların iniş pistini kullanım sırasını içeren öncelikli kuyruk "inis_pisti_kullanim_sirasi" öncelikli kuyruğudur. Uçak öncelik ve iniş saati bilgisi dikkate alınarak hazırlanmıştır.

Havalimanından kalkış yapacak uçakların kalkış pistini kullanım sırasını içeren öncelikli kuyruk "kalkis_pisti_kullanim_sirasi" öncelikli kuyruğudur. Uçak öncelik ve kalkış saati bilgisine göre hazırlanmıştır.

III. YÖNTEM

İlk olarak bağlı liste veri yapısı kullanılarak oluşturulacak öncelikli kuyruk yapısı için gerekli iki adet "struct" yapısı oluşturulmuştur. Bu "struct" yapıları "inis_pisti_kullanim_sirasi" ve "kalkis_pisti_kullanim_sirasi" şeklinde isimlendirilmiştir. "inis_pisti_kullanim_sirasi" isimli struct yapısı içerisinde:

- "int" tipinde; uçakların öncelik ID'sini tutan "o_id" değişkeni,
- Uçakların ID'sini tutan "u_id" değişkeni,
- Uçakların iniş saatini tutan "i_sa" değişkeni,
- Uçakların gecikme sürelerini tutan "gecikme" değişkeni,
- "struct" tipinde bir sonraki düğümü gösterecek "*sD" işaretçisi tanımlanmıştır.

"kalkis_pisti_kullanim_sirasi" isimli struct yapısında ise bu değişkenlere ek olarak "int" tipinde talep edilen iniş saatini tutan "t_i_sa" değişkeni ve kalkış saatini tutan "k_sa" değişkeni tanımlanmıştır.

A. Öncelikli Kuyruk Yapısı İçin Gerekli Fonksiyonların Oluşturulması

Bağlı listeler kullanılarak oluşturulacak olan öncelikli kuyruk yapısı için gerekli fonksiyonlar hazırlanmıştır. Bu fonksiyonlar:

- İstenilen verileri tutan düğümü kuyruğun sonuna ekleyen 4 parametrelili "sonaEkle" "void" fonksiyonu,
- İstenilen verileri tutan düğümü kuyruğun başına ekleyen 4 parametrelili "basaEkle" "void" fonksiyonu,
- İstenilen verileri tutan düğümü kuyruğun istenilen yerine ekleyen 5 parametrelili "arayaEkle" "void" fonksiyonu,
- Kuyruğun başındaki düğümü silen "bastanSil" "void" fonksiyonu,
- Kuyruğun sonundaki düğümü silen "sondanSil" "void" fonksiyonu,
- İstenilen bir sıradaki düğümü silen "aradanSil" "void" fonksiyonu,
- Kuyruğun boş veya dolu olduğunu belirten "bosMu" "int" fonksiyonu,
- Kuyruktaki düğüm değerlerini ekrana yazdıran "yazdir" "void" fonksiyonudur.

Bu fonksiyonlara ek olarak "kalkis_pisti_kullanim_sirasi" öncelikli kuyruğu için gerekli olan fonksiyonlar da aynı ismin başına "k_" eki eklenerek oluşturulmuştur.

B. "input.txt" Dosyasından Verilerin Okunması

"fopen()" fonksiyonu ile okuma modunda açılan "input.txt" dosyasından "fscanf()" fonksiyonu ile okuma yapılacaktır. "input.txt" dosyasından okunulması istenilen sayısal verilerden önce bulunan "oncelik_id", "ucak_id", "talep_edilen_inis_saati" kelime toplulukları üç defa art arda "fscanf()" fonksiyonu kullanılarak atlanmıştır. Ardından "while()" döngüsü içerisinde "fscanf()" fonksiyonu ile sıradaki

üç sayı okunup, "main()" fonksiyonu içerisinde oluşturulan "int" tipindeki "o_id", "u_id" ve "i_sa" değişkenlerine atanmıştır.

Havalimanı uçuş yönetim sistemi için gerekli işlemler yapıldıktan sonra ekranda görüntülenebilmesi için mevcut "while()" döngüsü sonuna "system("pause");" ifadesi, bir sonraki veriler okunmadan önce ekrandaki yazıların temizlenmesi için de "system("cls");" ifadesi kullanılmıştır.

C. İlk Uçak Verisinin Okunması

İlk veri okunmadan önce "inis_pisti_kullanim_sirasi" öncelikli kuyruğunda gerekli sıralama işlemlerinin yapılabilmesi için tüm değerleri "-100" ve "100" olan iki adet düğüm "sonaEkle()" fonksiyonu ile art arda öncelikli kuyruğa eklenmiştir. Öncelikli kuyruksa bulunabilecek maksimum düğüm sayısı 26 adettir. 24 adet uçak verisi ve kontrol amaçlı eklenen 2 adet düğüm dışında ekleme yapılamaz.

"input.txt" dosyasından ilk uçak verisi okunduktan sonra "bosMu()" fonksiyonu ile "inis_pisti_kullanim_sirasi" öncelikli kuyruğunda önceden eklenmiş iki adet düğüm dışında başka düğüm olmadığı teyit edilir ve okunan veriler "arayaEkle()" fonksiyonu ile halihazırda bulunan iki düğüm arasına eklenir.

D. Okunan Uçak Verilerinin Öncelikli Kuyruksa Uygun Yerlere Ekleneceği

"input.txt" dosyasından ilk uçak verisi okunup öncelikli kuyruğa eklendikten sonra sıradaki uçak verisi dosyadan okunur. Öncelikli kuyruğun azami kapasitesi olan 26 adet düğüme henüz ulaşılmadığı kontrol edildikten sonra okunan uçak verisinin "i_sa" değişkeninde tutulan "talep edilen inis saati" değerinin kuyruksa bulunan düğümlerdeki verilerle çakışıp çakışmadığı kontrol edilir. Eğer çakışma yoksa "arayaEkle()" fonksiyonu ile yeni uçak verisi öncelikli kuyruksa uygun yere eklenir.

Eğer çakışma var ise uçak yeni kontrollere maruz bırakılır. Oluşabilecek üç adet durum vardır. Bu durumlar; yeni eklenecek uçağın, kuyruksa inis saati çakışan uçaktan daha öncelikli olması (ilk durum olsun), kuyruksa inis saati çakışan uçağın yeni uçaktan daha öncelikli olması (ikinci durum olsun) ve her iki uçağın da eşit önceliğe sahip olmasıdır (üçüncü durum olsun). Her durum için farklı işlemler yapılır.

İlk durumda kuyruksa mevcut uçağın azami gecikme süresi olan 3 saat ertelenip ertelenmediği kontrol edilir. Eğer azami gecikme süresine ulaşmadıysa yeni uçağın indirilmesi ve eski uçağın ertelenmesi için "yui()" fonksiyonu çağırılır. Eğer eski uçak azami gecikme süresine ulaşıtıysa yeni uçağın ertelenmesi için "yue()" fonksiyonu çağırılır. İkinci durumda

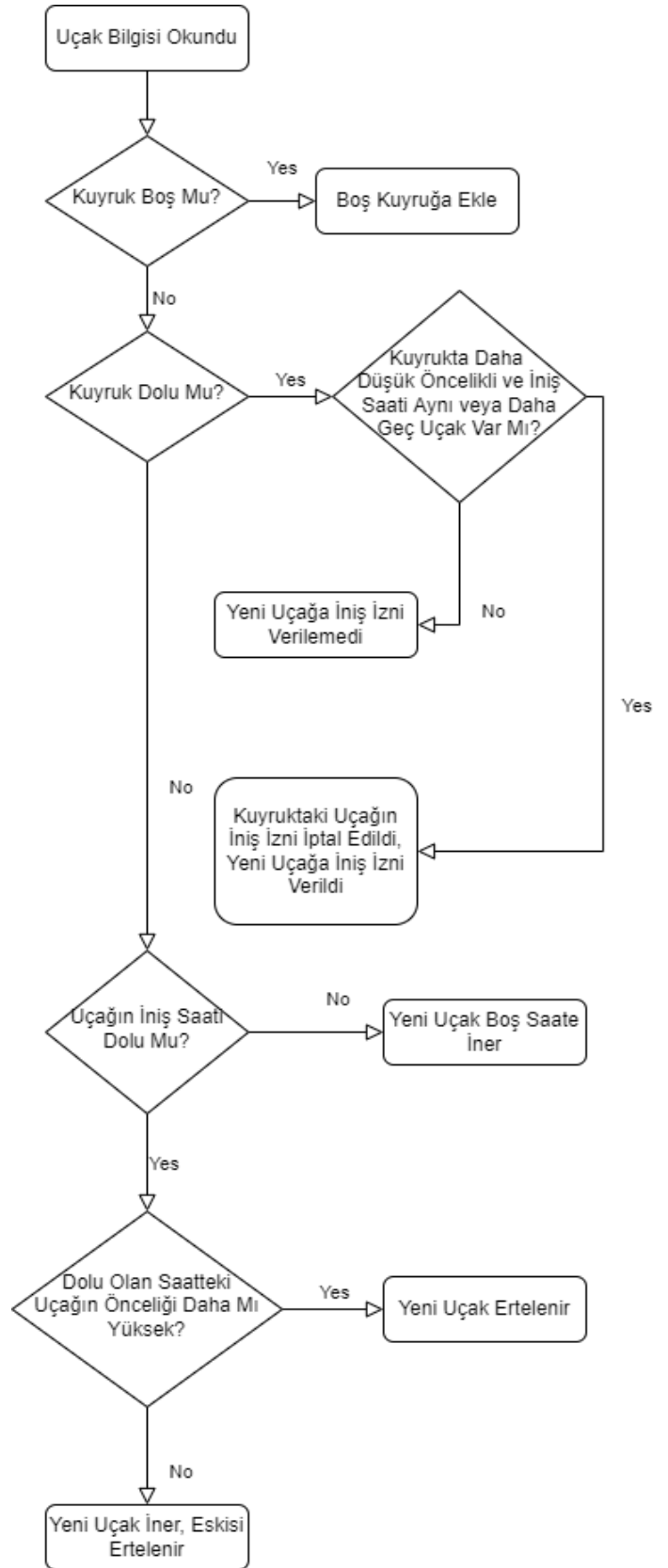


Fig. 1. İnş İzni Kontrolü Akış Şeması

ise yeni uçağın ertelenmesi için "yue()" fonksiyonu çağırılır. Üçüncü durumda ise iniş saati çakışan uçakların önceliği uçakların ID'sine göre karar verilir. ID sayısı düşük olan uçak daha önceliklidir. İlk iki durumdaki gibi gerekli kontroller yapılır ve gerekli fonksiyonlar çağırılır.

E. "yue()" Fonksiyonu

Yeni eklenecek uçakların ertelenmesi için oluşturulan bu fonksiyonda eklenecek uçak verisi en fazla 3 kere tekrar edebilecek bir döngüye girer. Her tekrarda kuyruktaki uçağın verisiyle yeni uçağın verileri karşılaştırılır. Karşılaştırma yapılırken öncelikle kuyruktaki mevcut uçağın azami gecikme süresine ulaşp ulaşmadığı kontrol edilir. Eğer uygun bir durum oluşmazsa yeni uçağa iniş izni verilemez ve Sabiha Gökçen Havalimanı'na yönlendirilir.

Uygun bir durum olduğunda ise "arayaEkle()" fonksiyonu ile yeni uçak gecikme ve yeni iniş saati bilgileriyle beraber kuyruktaki yerine eklenir ve eski uçağın bilgileri oluşturulan geçici değişkenlere atandıktan sonra ertelenebilmesi için "ee()" fonksiyonu çağırılır.

F. "yui()" Fonksiyonu

Yeni uçağın gecikmesiz bir şekilde kuyruğa eklenebilmesi için oluşturulan bu fonksiyonda yeni uçak "arayaEkle()" fonksiyonu ile kuyruğa eklenir. Kuyrukta bulunan mevcut uçağın bilgileri oluşturulan geçici değişkenlere atandıktan sonra "aradanSil()" fonksiyonu ile kuyruktan silinir ve ertelenmesi için "ee()" fonksiyonu çağırılır.

G. "ee()" Fonksiyonu

Halihazırda kuyrukta bulunan uçakların ertelenebilmesi için oluşturulan bu fonksiyonda öncelikle uçağın gecikme değeri bir artırılır. Yeni gecikme değeri 3'e ulaştıysa yeni iniş saatinin olduğu saatte pistin dolu olup olmadığı kontrol edilir. Eğer boşsa eski uçağın iniş bilgileri yeni gecikme ve iniş saati bilgileriyle "arayaEkle()" fonksiyonu ile kuyruğa tekrar eklenir. Eğer doluysa da kuyruktaki iniş saati çakışan uçağın azami gecikme süresine ulaşp ulaşmadığı kontrol edilir. Ulaşmadıysa eski uçak, diğer eski uçağın yerine geçer ve diğer eski uçağın ertelenmesi için "ee()" fonksiyonu tekrardan çağırılır. Eğer uçak azami gecikme süresine ulaştıysa ertelenmek istenen eski uçağın izni iptal edilir.

Mevcutta bulunan uçak ertelenirken gecikme değeri arttırdıktan sonra 3'e ulaşmamışsa uçak bir döngüye girer. Bu döngüde kuyrukta uygun yer bulma ve erteleme işlemleri yapılır. Uçak boş saat bulabilmişse kuyruktaki uygun boşluğa eklenir. Eğer kuyruktaki başka bir uçağın yerini alırsa, kuyruktaki uçağın da ertelenmesi için "ee()" fonksiyonu tekrar çağırılır.

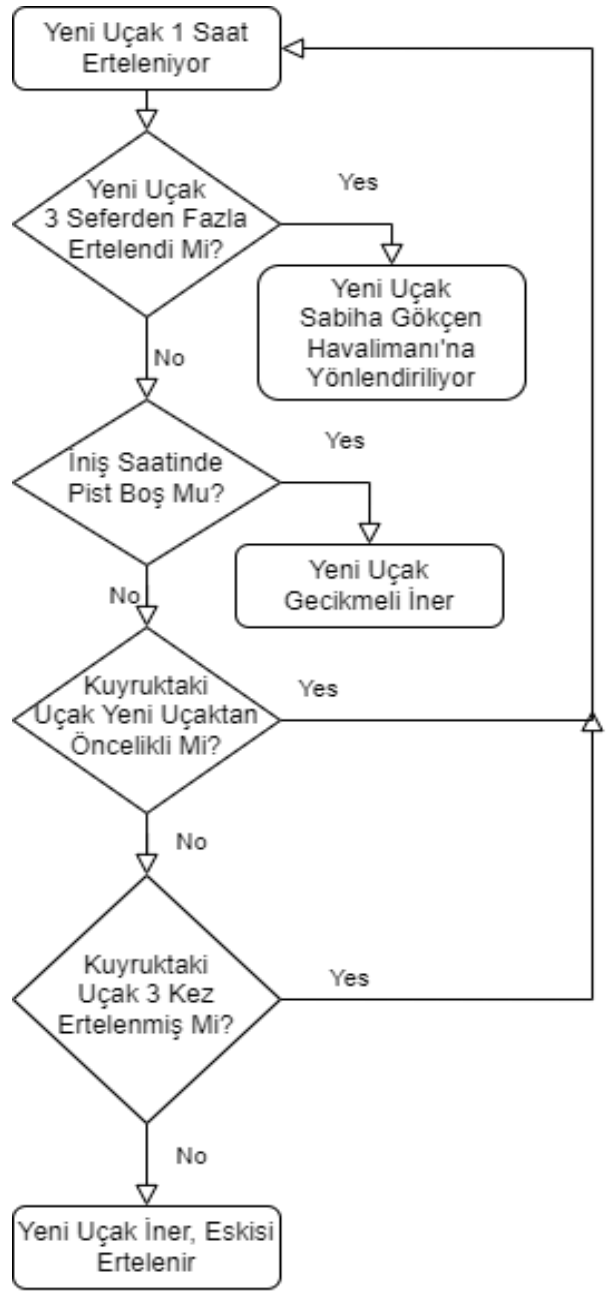


Fig. 2. Yeni Uçağın Ertelenmesi Akış Şeması

H. Kuyruk Kapasitesinin Dolması Durumu

Yeni bir uçak eklenirken kuyruk kapasitesi dolmuşsa, kuyrukta "Önceliği yeni uçaktan düşük ve iniş saati yeni uçağın iniş saatine eşit veya daha geç saatte olan uçak var mı?" diye kontrol edilir. Eğer varsa yeni uçak eski uçağın yerini alır ve eski uçağın iniş izni iptal edilir. Eğer yoksa yeni uçağa iniş izni verilemez ve Sabiha Gökçen Havalimanı'na yönlendirilir.

I. "guncel()" Fonksiyonu

"input.txt" dosyasından ilk uçak bilgileri okunduktan sonra her bir okuma öncesi ve tüm okumalar bittikten sonra "guncel()" fonksiyonu çağırılır. Bu fonksiyonda "kalkis_pisti_kullanim_sirasi" kuyruğundaki tüm elemanlar silinir. Ardından "inis_pisti_kullanim_sirasi" kuyruğundaki ilk ve son düğümler hariç tüm düğümler "kalkis_pisti_kullanim_sirasi" öncelikli kuyruğuna da eklenir.

"output.txt" dosyası yazma modunda açılır. Bu dosyanın ilk satırına "oncelik_id ucak_id talep_edilen_inis_saati inis_saati gecikme_suresi kalkis_saati" kelime grubu yazdırılır. Sonrasında "kalkis_pisti_kullanim_sirasi" öncelikli kuyruğundaki düğümlerde saklanan uçak verileri "output.txt" dosyasına yazdırılır ve dosya kapatılır. "k_yazdir()" fonksiyonu da çağırılarak "kalkis_pisti_kullanim_sirasi" öncelikli kuyruğundaki veriler ekrana yazdırılır.

IV. DENEYSEL SONUÇLAR

Öncelikli kuyrukların temel fonksiyonları geliştirilerek proje amacına uygun olacak şekilde düzenlendi.

Yeni uçakların gelmesi ve eski uçakların ertelenmesinden dolayı koşul kontrolleri sonsuz döngüye giriyordu. Koşul kontrollerini fonksiyon içerisinde yapıp, o fonksiyonların içerisinde de aynı fonksiyonu tekrardan çağırarak bu sonsuz döngüden çıkıldı.

Azami gecikme süresine ulaşan uçaklar tekrardan ertelenmeye çalışıldığında, eski uçak kuyruktan siliniyor ve uçak verisi kayboluyordu. Azami gecikme süresine ulaşan uçakları en yüksek öncelikli uçak kategorisine uygun kontrollerden geçirerek 3'ten fazla ertelenmemesi sağlandı.

V. SONUÇ

Bağlı listeler ve öncelikli kuyruklar kullanılarak "Havalimanı Uçuş Yönetim Sistemi" hazırlandı. Alınan veriler belirli koşullara ve önceliklerine göre sıralandı. Çeşitli fonksiyonlar kullanılarak karışıklığın azaltılması sağlandı.

KAYNAKÇA

- [1] medium.com. (2019). Doğrusal Veri Yapılar 4, Erişim Tarihi: 10.12.2021, <https://medium.com/@tolgahan.cepel/dogrusal-veri-yapilari-4-kuyruk-queue-dcbd07e8ba77/>
- [2] cdn-acikogretim.istanbul.edu.tr. KUYRUK (QUEUE) VERİ YAPISI, Erişim Tarihi: 10.12.2021, https://cdn-acikogretim.istanbul.edu.tr/auzefcontent/20_21_Guz/veri_yapilari/7/
- [3] yazilimtuneli.com. (2019). C Programlama Yığın ve Kuyruk Örnek, Erişim Tarihi: 10.12.2021, <http://www.yazilimtuneli.com/2019/11/c-programlama-ygn-ve-kuyruk-ornek/>
- [4] Yazılıma Gönül Ver. (2019). C++ Priority Queue (Öncelikli Kuyruk) Sınıfı ve Methodları, Erişim Tarihi: 10.12.2021, <https://www.youtube.com/watch?v=IqYh57D67-g/>
- [5] diagrams.net. Akış Şeması, Erişim Tarihi: 12.12.2021, <https://app.diagrams.net/>