

Kocaeli Üniversitesi Programlama Laboratuvarı Proje

3

Halime Nur Yalçın-210201108 Edanur Çevüt-190201035

20 December 2021

1 Özet

Bu doküman Programlama Laboratuvarı 1 dersi 3. Projesi olan Havalimanı Uçuş Yönetim Sistemi için hazırlanmıştır. Dokümanda projenin tanımı, çözüme yönelik yapılan araştırmalar, kullanılan yöntemler, proje hazırlanırken kullanılan geliştirme ortamı ve kod bilgisi gibi programın oluşumunu açıklayan başlıklar bulunmaktadır. En sonda da kaynakça kısmı bulunmaktadır.

2 Proje Tanımı

Projenin adı Havalimanı Uçuş Yönetim Sistemi'dir. Projenin amacı öncelikli kuyruk (priority queue) kullanarak bir havalimanı uçuş yönetim sistemi oluşturmaktır. Öncelik sırası, iniş saati ve 3'ten fazla kez ertelenme durumunda farklı havalimanına gönderilme gibi durumlar mevcuttur. Bu problemlerin çözümü için çözüm kuyruk veri yapısı olmuştur. Kuyruk veri yapısı günlük hatta en çok karşılaşılan bir yapıdır. Üniversitesinin yemekhanesinde öğrencilerin yemek almak için oluşturdukları kuyruk, markette satın alınan ürünlerin üc-

retini ödemek için müşterilerin kasaların önünde oluşturdukları kuyruk, hastanede hastaların muayene olmak için doktorun odası önünde oluşturdukları kuyruk bu yapıya verilecek iyi örneklerdendir. Bu yapıların ortak özelliği, sırada bekleyen kişiler için sunulan hizmete erişebilecek tek bir nokta bulunmasıdır. Kuyruk yapısını tam olarak anlayabilmek için doktorun odası önünde bekleyen hastaları düşünelim. Sıraya ilk giren hasta (Kuyruğun önündeki hasta), ilk muayene olacak kişidir ve o muayene olduktan sonra sıra bir sonraki hastaya gelir. Bu esnada muayene olmak için hastaneye yeni gelen bir hasta olduğunda bu hasta kuyruğun en sonuna geçmek zorundadır. Muayene olmak için doktorun odası önünde bekleyen hastalar düşünüldüğünde en son muayene olacak hasta kuyruğa en son eklenen hasta olacağını tahmin etmek zor değildir. Bazı problemlerin çözümünde doğrudan kuyruk oluşturulamaz. Örneğin; bir hastanede muayene sırasına girmiş insanlar arasında durumu acil olan birisi bulunabilir ve bu kişi muayene için öncelikli hale gelebilir. Bu gibi durumlarda öncelikli kuyruk kullanılır. Öncelikli kuyrukta ilk giren ilk çıkar mantığı geçerli değildir, önemli olan önceliklidir.

3 Arařtırmalar ve Yöntem

Arařtırmalar daha çok kuyruk veriyapısı üzerine oldu. Elde ettiğimiz sonuçları řu řekilde aktarabiliriz: Kuyruk veri yapısının tanımı, “Eleman ekleme işleminin son-
dan (rear), eleman çıkarma işleminin baş-
tan (front) yapıldığı doğrusal bir veri ya-
pısıdır.” řeklinde verilebilir. Başka bir ifa-
deyle kuyruk veri yapıları, önce gelen ele-
mana ilk hizmetin verildiği veri yapılarıdır.
Yığın veri yapısından farklı olarak ilk ge-
len – ilk çıkar (First In First Out – FIFO)
prensibine göre çalışır. Literatürde Bir ele-
manın kuyruğa girmesi enqueue, bir elema-
nın kuyruktan silinmesi dequeue olarak kul-
lanılır. Kuyruk veri yapısında enqueue iş-
lemleri kuyruğun arkasından, dequeue iş-
lemleri kuyruğun önünden yapılır. Kuyruk
veri yapısında dolu bir kuyruğa eleman ek-
lemek overflow hatası döndürürken boş bir
kuyruktan eleman silmek underflow hatası
döndürür. Kuyruk yapıları da yığınlara ben-
zer biçimde diziler ve bağı listeler kulla-
nılarak oluşturulup çalıştırılabilir. Kavram
basit olsa da, bir kuyruğu programlamak,
bir yığını programlamak kadar basit değıl-
dir. Hastanede muayene olmak için bekle-
yen hasta örneğine geri dönelim ve bir has-
tanın muayene olup kuyruktan ayrıldığını
varsayalım. Bu durumda sıradaki herkes bir
öne hareket etmek zorundadır, Şimdi, bir
seferde yalnızca bir kişinin hareket edebil-
diğini varsayalım. Böylece, ikinci kişi birinci
kişinin bıraktığı boşluğu doldurmak için öne
çıkarak ve ardından üçüncü kişi, ikinci kişinin
bıraktığı boşluğu doldurmak için öne çıkar
ve bu böyle devam eder. Şimdi, herkes öne
çıkana kadar hiç kimsenin sıraya ekleneme-

yeceğini düşünelim. Çizginin çok yavaş ha-
reket edeceğini görebilirsiniz.

Bu projede struct yapısı ile bize verilen
input dosyasındaki öncelik id, uçak id, talep
edilen inis saati ve gerekli olan ertelenme sa-
yisi bilgilerinin bir isim altında toplanma-
sını sağladık. Fonksiyonlarımız inis cagir,
kalkis ayarla, kayit sil, dosyayi aktar, kontrol
ve yazdir'dir. İnis cagir fonksiyonuyla kont-
rol fonksiyonunu çağırdık, "dosyayi aktar"
fonksiyonu ile verilen input.txt dosyasını sa-
dece okuma yapacağımız için "r" ile okut-
tuk. If else yapısı ile dosyanın boş olup ol-
madığını kontrol ettirip eğer doluysa while
döngüsüyle yine if else yapısını kullanıp
kuyruktaki sıranın doluluğuna göre eleman-
ları yazdırdık. "kontrol"fonksiyonuyla kapa-
sitenin doluluğunu kontrol edip kuyruk sıra-
sına göre fonksiyonu devam ettirdik. Bizden
istenen maksimum 3 kez ertelenme kuralını
öncelikle saat kontrolü ve ardından aynı sa-
atteki uçakların öncelik no kontrollerini ya-
parak gerçekleřtirdik. Kalkis ayarla fonksi-
yonuyla output.txt dosyamızı "w" ile yaz-
dırdık. Kayit sil fonksiyonuyla Sabiha Gök-
çen Havalimanına aktarılan uçaklar kuy-
ruktan silindi. Ardından öncelik sıralama-
sında 3'ten fazla kez ertelenme yaşayan
uçakların sayısını belirleyip Sabiha Gök-
çen Havalimanına'na yönlendirilmesini sağ-
ladık. "yazdir" fonksiyonuyla oluşan kuyru-
ğun son hali ekrana çıktı olarak verildi.

4 Geliřtirme Ortamı

Projeyi Windows sistemde, Dev-C++
üzerinde geliřtirip yine Dev-C++ ve Co-
deBlocks kullanarak derledik.

5 Kod Bilgisi

5.0.1 Algoritme Şeması

Şemalar dosya içinde mevcut.

5.1 Algoritma

Bu projede struct yapısı ile bize verilen input dosyasındaki öncelik id, uçak id, talep edilen inis saati ve gerekli olan ertelenme sayısı bilgilerinin bir isim altında toplanmasını sağladık. Fonksiyonlarımız inis çağır, kalkis ayarla, kayıt sil, dosyayı aktar, kontrol ve yazdır'dır. İnis çağır fonksiyonuyla kontrol fonksiyonunu çağırdık, "dosyayı aktar" fonksiyonu ile verilen input.txt dosyasını sadece okuma yapacağımız için "r" ile okuttuk. If else yapısı ile dosyanın boş olup olmadığını kontrol ettirip eğer doluysa while döngüsüyle yine if else yapısını kullanarak kuyruktaki sıranın doluluğuna göre elemanları yazdırdık. "kontrol"fonksiyonuyla kapasitenin doluluğunu kontrol edip kuyruk sırasına göre fonksiyonu devam ettirdik. Bizden istenen maksimum 3 kez ertelenme kuralını öncelikle saat kontrolü ve ardından aynı saatteki uçakların öncelik no kontrollerini yapılarak gerçekleştirdik. Kalkis ayarla fonksiyonuyla output.txt dosyamızı "w" ile yazdırdık. Kayıt sil fonksiyonuyla Sabiha Gökçen Havalimanına aktarılan uçaklar kuyruktan silindi. Ardından öncelik sıralamasında 3'ten fazla kez ertelenme yaşayan uçakların sayısını belirleyip Sabiha Gökçen Havalimanına'na yönlendirilmesini sağladık. "yazdır" fonksiyonuyla oluşan kuyruğun son hali ekrana çıktı olarak verildi.

5.1.1 İstatistik

Program kodu 150 satırlık dosyadan oluşmaktadır. Kod düzenini sağlamak için yaklaşık 8 boş satır ve 5 yorum satırı kullanılmıştır. Kullandığımız kütüphaneler ve ne için kullandığımız kabaca aşağıdaki gibidir: <stdio.h> Çıktı ve girdi almak için

5.1.2 Programın Derlenmesi

Programın kaynak kodu tek dosyadan oluşmaktadır. Bu dosyayı Dev-C++ ve CodeBlocks ile derleyebilirsiniz.

6 Kaynakça

<http://www.baskent.edu.tr/~tkaracay/etudio/ders/prg/dataStructures/Collections/ClassPr>
<https://medium.com/@tolgahan.cepel/dohttps://docplay>
[Algoritma-gelistirme-ve-veri-yapilari-8-kuyruk-ve-yigin-yapisi-mustafa-kemal-universitesi.html](https://nerdbook.wordpress.com/2018/03/26/kuyruk-veri-yapisi-queue/)
<https://nerdbook.wordpress.com/2018/03/26/kuyruk-veri-yapisi-queue/>
<https://www.youtube.com/watch?v=1hKLn12poa8>
<https://www.youtube.com/watch?v=mCyaYxEhsY0>
<https://www.youtube.com/watch?v=1hKLn12poa8>
<https://ubenzer.com/2009/01/cde-oncelik-kuyruğu/>
<https://docs.microsoft.com/tr-tr/azure/architecture/patterns/priority-queue>