

BLM209 PROGRAMLAMA LABORATUVARI I

PROJE 3

Bariş Kakilli

Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

200201012@kocaeli.edu.tr

Muhammed Sina Çimen

Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

200201032@kocaeli.edu.tr

I. ÖZET

Bu projede amaç öncelikli kuyruk (priority queue) yapısını kullanarak bir havalimanı uçakların geliş saatine göre ve önceliklerine göre uçuş yönetim sistemi oluşturmak ve verilen ilgili input dosyasına göre bir sonuç oluşturup bunları output dosyasında göstermektir.

II. GİRİŞ

Kuyruk (Queue):

Kuyruk, ilk giren eleman ilk çıkar (First In First Out – FIFO) mantığında çalışan bir veri yapısıdır. Örneğin, kuyruk veri yapısı bankada işlem yaptırmak için sıraya girmiş insanlara benzetilebilir. Sıraya ilk giren kişi ilk işlem yaptıracaktır. Kuyruk tasarımı için dizi ya da bağlı liste kullanılabilir. Dizi kullanılan sabit boyutlu, bağlı liste kullanılarak değişken boyutlu kuyruk oluşturulabilir. Kuyrukta işlemler iki uçtan yapılır. Kuyruk veri yapısında yapılabilecek işlemlerden bazıları aşağıdaki gibidir:

- enqueue(): Kuyruğun önüne eleman ekler.
- dequeue(): Kuyruğun sonundan eleman çıkarır.
- peek(): Silme işlemi uygulamadan sıradaki elemanı (front işaretçisinin gösterdiği düğüm) döndürür.

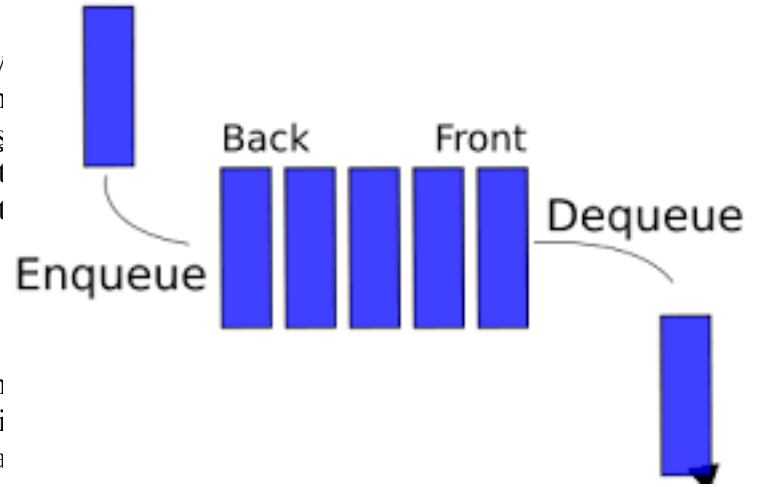


Fig. 1. genel que yapısı

Öncelikli Kuyruk (Priority Queue):

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. Öncelikli kuyruk veri yapısında yapılabilecek işlemlerden bazıları aşağıdaki gibidir: - add(): Kuyruğa eleman eklemek için kullanılır. - poll(): Kuyruktaki son elemanı döndürür ve elemanı kuyruktan siler. - peek(): Kuyruktaki son elemanı silmeden döndürür. - clear(): Kuyruktaki bütün elemanları siler. - remove(): Kuyruktaki belirtilen elemanı siler.

Proje kapsamında bizden öncelikli kuyruk yapısını kullanarak 1 iniş 1 kalkış olmak üzere iki pisti

bulunan ve bir günlük uçuş yönetimini yaptığımız bir havalimanı tasarlamamız istenmiştir. Bu havalimanında aynı anda sadece 1 uçak kalkış yapabiliyorken sadece 1 uçak iniş yapabilmektedir. Uçakların her biri iniş ve kalkışta farklı önceliklere sahiptir ve 1 günde maksimum 24 uçak iniş için izin isteyebilmektedir. Havalimanındaki uçakların öncelik sırası, iniş saati, gecikme süresi ve kalkış saati bilgileri kullanılarak; iniş pistini ve kalkış pistini kullanım sırasının belirlenmesi istenmektedir. Havalimanına iniş yapacak uçaklar öncelikle kuleden iniş yapabilmek için izin talep edecektir. İniş izni talep eden her bir uçak için havalimanında yeterli kapasite olup olmadığı yani kuyrukta yeni uçak eklemek için yer var mı kontrol edilmelidir. Kuleden iniş izni talep eden uçaklar için öncelikle, iniş talep edilen saatte başka bir uçak iniş talep ediyor mu kontrol edilmelidir. Yoksa uçak indirilir fakat varsa aynı vakitte gelen uçaklar için öncelik sırasına göre iniş izni verilir. İniş izni talep eden her uçak için “İniş izin talebiniz onaylanmıştır” veya “... nedeniyle iniş izni verilememektedir.” şeklinde ekranda yazdırılacaktır. 4 çeşit uçak vardır ve öncelikleri yüksekte düşüğe ambulans uçağı, savaş uçağı, yolcu uçağı, kargo uçağı şeklindedir. Havalimanına iniş talep eden uçakların önceliği, uçak numarası ve talep ettiği iniş saati bize proje kapsamında verilen input.txt dosyasından okunacaktır. Uçakların iniş kalkış süreleri ise yok sayılacak ve hesaplama dâhil edilmeyecektir. Havalimanına iniş yapan her uçağın, kalkış için bekleme süresi 1 saattir. Uçakların kalkış saatine, ötelenmeden dolayı oluşan gecikme süreleri dâhil edilecek ve kalkış saati bu bilgiler göz önünde bulundurularak hesaplanacaktır. Önceliği yüksek olan uçaklar nedeniyle önceliği düşük olan herhangi bir uçağın inişi, maksimum 3 kez ertelenebilir. Eğer 3’ten fazla ertelenme durumu söz konusuysa, öncelik gözetilmeksizin beklemede olan uçağın kalkışı gerçekleştirilecektir. Kuleden bir günde maksimum 24 uçak iniş için izin talep edebilir. Eğer bu kapasite dolmuşsa; iniş için onay alan uçaklardan en az birinin önceliği (X uçağı olsun), iniş izni onayı bekleyen uçağın (Y uçağı olsun) önceliğinden düşükse; yüksek öncelikli yeni uçağa (Y) iniş onayı verilir. Daha önce onay almış ve önceliği düşük olan uçak (X) başka bir havalimanına yönlendirilecektir. İniş izni daha önceden onaylanan uçağın (X) izni iptal edilmişse; “Acil iniş

yapması gereken ... (Y) uçağı nedeniyle iniş izniniz iptal edilmiştir, iniş için Sabiha Gökçen Havalimanı’na yönlendiriliyorsunuz.” şeklinde ekranda yazdırılacaktır. İniş izni talep eden uçakların her biri satır satır input.txt’den okunacak ve okunan her bir satır ekranda gösterilecektir. Her yeni input satırı okunduğunda, kalkış yapacak olan uçakların bulunduğu output.txt dosyası güncellenmeli ve güncel öncelikli kuyruk ekranda gösterilecektir. Bu isterler sonucunda her satırında öncelik id, uçak id, talep edilen iniş saati, iniş saati, gecikme süresi ve kalkış saati olmak üzere toplamda 6 bilgi içeren bir output.txt oluşturulacaktır.

III. YÖNTEM

Program C programlama dilinde yazılmış olup, derleyici olarak “GCC”, IDE olarak “CodeBlocks” kullanılmıştır. Programın çalışabilirliği “Windows 10 64 bit” işletim sisteminde test edilmiştir. Proje bir ekip tarafından geliştirildiğinden dolayı işlevlerin birleşimi, kodun takibi gibi amaçlarla “Github” versiyon kontrol sistemi kullanılmıştır. Projede temel veri yapısı olarak öncelikli kuyruk yapısı kullanılmıştır.

Projede ilk başta input.txt dosyamızı satır satır okuyan ve verileri ayırt eden bir kod parçası yazdık. Bu kod parçasının hemen ardından verilerimizi sıralamayı amaçladık. Bu sıralama algoritmasını ise üç defa çağırdık. Çünkü ilk sıralamada iniş talep saatini esas aldık. İkinci sıralamada iniş önceliğine göre kontrol ettik ve son olarak da bu durum da eşit olursa diye uçak idlerini kıyaslayan bir sıralama kodu daha ekledik.

Bu sıralama işlemlerinden sonra txt’yi arraye aktardık sonra eleman sayısı kadar bir döngüye girip döngüde saat 1’den başlayıp saat 24’e kadar uçakları indirdik. bu döngü içinde ilk olarak o saat içinde kaç uçak izin istiyor onu bulduk sonrasında ise cezalı(3ten fazla bekleyen var mı onu hesaplıyoruz eğer varsa ceza_flag değişkenini 1 yapıyoruz) var mı yok mu onu hesaplıyoruz hesapladıktan sonra her beklemede olan uçağa +1 ceza yazıyoruz. yazma kısmında cezalı mı değil mi ceza_flag değişkeni ile bakıyoruz eğer cezalı değil ise inmek isteyen uçakların önceliği en düşük olanı hesaplıyoruz eğer 1 tane en düşük öncelikli varsa onu hemen piste indiriyoruz(que)

eğer birden fazla düşük öncelikli varsa bunlar arasında id karşılaştırması yapıyor ve en düşük olanı indiriyoruz. en son kısımda saati 1 arttırıyoruz tekrardan. yazma kısmının cezalılı yerinde önceliğe yine bakıyoruz ama bu sefer ceza_sayisi 3 yaparak bakıyoruz böylece sadece bekleyen uçakları karşılaştırmış oluyoruz eğer 1 uçak beklemedeyse o uçak hemen iniyor birden fazla ise öncelik ve idsi daha az olan Sabiha Gökçen Havalimanına aktarılıyor.

IV. DENEYSEL SONUÇLAR

```
1 1 14
2 2 13
2 3 11
3 4 10
4 5 16
2 6 10
3 7 2
3 8 14
4 9 17
4 10 13
1 11 1
1 12 1
2 13 9
3 14 16
3 15 13
2 16 20
3 17 1
3 18 24
1 19 23
4 20 21
3 21 19
4 22 1
2 23 2
3 24 8
1 25 16
4 26 20
2 27 2
4 28 3
```

Fig. 2. örnek input

V. SONUÇ

Projede bizden istenen tüm isterler eksiksiz bir şekilde başarıyla tamamlanmıştır.

```
1 11 1 1 0 2
1 12 1 2 1 3
2 23 2 3 1 4
3 17 1 4 3 5
2 27 2 5 3 6
4 28 3 6 3 7
3 24 8 8 0 9
2 13 9 9 0 10
2 6 10 10 0 11
2 3 11 11 0 12
3 4 10 12 2 13
2 2 13 13 0 14
1 1 14 14 0 15
3 8 14 15 1 16
3 15 13 16 3 17
1 25 16 17 1 18
3 14 16 18 2 19
4 5 16 19 3 20
4 9 17 20 3 21
2 16 20 21 1 22
3 21 19 22 3 23
4 26 20 23 3 0
4 20 21 0 3 1
1 19 23 1 2 2
3 18 24 2 2 3
```

Fig. 3. örnek output

```
saat su an: 1 izin isteyen uçak sayısı: 4
sırası gelen 11 id'li uçak inis yapıyor

saat su an: 2 izin isteyen uçak sayısı: 6
sırası gelen 12 id'li uçak inis yapıyor

saat su an: 3 izin isteyen uçak sayısı: 6
sırası gelen 23 id'li uçak inis yapıyor

saat su an: 4 izin isteyen uçak sayısı: 5
bekleme yapan 17 id'li uçak inis yapıyor
birden fazla uçak beklemede olduğu için bu uçaklar sabiha gokcene aktarılıyor:
22 idli uçak sabiha gokcene aktarılıyor

saat su an: 5 izin isteyen uçak sayısı: 3
bekleme yapan 27 id'li uçak inis yapıyor
birden fazla uçak beklemede olduğu için bu uçaklar sabiha gokcene aktarılıyor:
7 idli uçak sabiha gokcene aktarılıyor
```

Fig. 4. örnek çıktı

VI. YALANCI KOD

0-Başla

1-input.txt'yi izin isimli yapı(struct) dizisine aktar
2-diziyi saat, öncelik ve idlerine göre sırala
3-döngü başlangıcı(eleman sayısı kadar)
4-saate göre kaç uçak izin istiyor hesapla
5-ceza(3ten fazla erteleme yaşayan uçaklar) değeri
3ten fazla olan var mı kontrol et
5a-varsa ceza_flag değerini 1 yap ve cezalı uçağın adresini tut
6-izin isteyen tüm uçakların ceza değerini arttır
7-eğer cezalı yoksa
7a-izin isteyen uçaklardan oncelik idsi en kucuk olanı ve kac adet mevcut olduğunu bul
7b-eğer 1 adet ise o uçağı kalkıs isimli kuyruk yapısına ekle
7c-1den fazlaysa bu uçaklar arasında ucak idsi en düşük olanı bul ve onu kalkıs isimli kuyruk yapısına ekle
8-eğer cezalı varsa
8a-cezali sayisi 1 adetse o uçağı kalkıs isimli kuyruk yapısına ekle
8b-cezali sayisi 1den fazlaysa izin isteyen cezalı uçaklardan oncelik idsi en kucuk olanı ve kac adet mevcut olduğunu bul
8c-eğer 1 adet ise o uçağı kalkıs isimli kuyruk yapısına ekle geri kalan cezalı uçakları izin isimli yapıdan çıkar
8d-1den fazlaysa bu uçaklar arasında ucak idsi en düşük olanı bul ve onu kalkıs isimli kuyruk yapısına ekle
8d-geri kalan cezalı uçakları izin isimli yapıdan çıkar
9-ceza_flag(cezalı olup olmadığını kontrol eden değer) değerini 0la
10-saati 1 arttır
11-eğer i değeri eleman sayısından az ise 3. adıma geri dön
12-kalkıs isimli kuyruk yapısının elemanlarını sırasıyla output.txtye aktar

VII. KAYNAKÇA

- [1] <https://stackoverflow.com/>
[2] <https://www.geeksforgeeks.org/>
[3] <http://www.baskent.edu.tr/tkaracay/etudio/ders/prg/dataStructures/Collections/ClassPriority>

[4] <https://medium.com/@tolgahan.cepel/do%C4%9Frusal-veri-yap%C4%B1lar%C4%B1-4-kuyruk-queue-dcbd07e8ba77>