

BLM209 PROGRAMLAMA

LABORATUVARI I

PROJE 3

PROJE TESLİM TARİHİ: 17.12.2021

PROJENİN ADI: Havalimanı Uçuş Yönetim Sistemi

PROJENİN AMACI: Öncelikli kuyruk (priority queue) kullanarak bir havalimanı uçuş yönetim sistemi oluşturmak.

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.

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 önceliktir. 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.

Kuyruk ve öncelikli kuyruk için daha fazla bilgiye aşağıdaki sitelerden erişilebilir:

- <http://www.baskent.edu.tr/~tkaracay/etudio/ders/prg/dataStructures/Collections/ClassPriorityQueue.pdf>

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

PROJENİN KONUSU:

1 iniş 1 kalkış olmak üzere 2 pisti bulunan İstanbul Havalimanı'nda gün içerisinde (1-24 saat dilimi boyunca) yapılan uçuşların yönetimi için bir sistem geliştirilecektir. 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 hedeflenmektedir.

- Havalimanına iniş yapacak uçaklar öncelikle kuleden iniş yapabilmek için izin talep etmelidir.
- İniş izni talep eden her bir uçak için havalimanında yeterli kapasite olup olmadığı kontrol edilmelidir (*inis_pisti_kullanim_sirasi* öncelikli kuyruğunda yeni uçak eklemek için boş alan var mı?).
- Kuleden iniş izni talep eden uçaklar için öncelikle, iniş talep edilen saatte pistin dolu mu boş mu olduğu kontrol edilmelidir. Pist boş ise iniş yapılmak istenen saate izin verilmeli ve *inis_pisti_kullanim_sirasi* 'nda uygun yere eklenmelidir. Aksi halde uçakların iniş sıralaması önceliğe göre belirlenmelidir.
- İniş izni talep eden her uçak için “İniş izin talebiniz onaylanmıştır” veya “... nedeniyle iniş izni verilememektedir.” şeklinde ekranda yazdırılmalıdır.
- Uçakların iniş ve kalkış saatleri önceliğe göre belirlenecektir. Uçakların öncelik (*oncelik_id*) sıralaması şu şekildedir (yüksekten düşüğe):

1. Ambulans uçağı
2. Savaş uçağı
3. Yolcu uçağı
4. Kargo uçağı

- Havalimanına iniş talep eden uçakların önceliği (*oncelik_id*), uçak numarası (*ucak_id*) ve talep ettiği iniş saati (*talep_edilen_inis_saati*) **input.txt** dosyasından okunacaktır.
- Tüm uçakların iniş ve kalkış süreleri eşittir ve hesaplamalara dâhil edilmeyecektir.
- Havalimanına iniş yapan her uçağın, kalkış için bekleme süresi 1 saattir. Uçakların kalkış saatine, ötelenmeden dolayı oluşan gecikme süreleri dâhil edilmelidir. Kalkış saati bu bilgiler göz önünde bulundurularak hesaplanmalıdır.
- Aynı önceliğe sahip iki uçak, aynı saatte kalkış yapacaksa öncelik ilk iniş yapan uçağa verilmelidir.
- Önceliği yüksek olan uçaklar nedeniyle önceliği düşük olan herhangi bir uçağın uçuşu, maksimum 3 kez ertelenebilir. Eğer 3'ten fazla ertelenme durumu söz konusuysa, öncelik gözetilmeksizin beklemede olan uçağın kalkışı gerçekleştirilmelidir.
- Kuleden bir günde maksimum 24 uçak iniş için izin talep edebilir. Eğer bu kapasite dolmuşsa;
 - İniş 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 yönlendirilmelidir.
 - 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 yazdırılmalıdır.
- İniş izni talep eden uçakların her biri satır satır **input.txt** 'den okunmalıdır. Okunan her bir satır ekranda gösterilmelidir.

- Her yeni input satırı okunduğunda, kalkış yapacak olan uçakların bulunduğu **output.txt** dosyası güncellenmeli ve güncel *kalkis_pisti_kullanim_sirasi* öncelikli kuyruğu ekranda gösterilmelidir.
- Projede kullanılacak **input.txt** dosyası ekte verilmiştir. **output.txt** dosyası ise proje içerisinde öğrenci tarafından oluşturulacaktır.
- **input.txt** dosyası aşağıdaki bilgileri içermektedir:
 - Havalimanına 1 günde iniş izni talep eden uçakların listesi
 - Dosyadan sırasıyla (satır satır) boşluk ile ayrılmış 3 bilgi (*oncelik_id, ucak_id, talep_edilen_inis_saati*) okunacaktır.
- **output.txt** dosyası uçakların kalkış saati bilgisine göre oluşturulmalıdır. Dosyanın içeriği şu şekilde olmalıdır:
 - Her satırda *oncelik_id, ucak_id, talep_edilen_inis_saati, inis_saati, gecikme_suresi, kalkis_saati* olmak üzere toplamda 6 bilgi içermelidir.
 - *inis_saati* bilgisi; eğer uçak talep ettiği saatte iniş yapmışsa *talep_edilen_inis_saati* ile aynı olmalıdır. Ancak talep edilen saatte önceliği yüksek başka bir uçak iniş yapacaksa belirlenen yeni iniş saati, kalkış saatinin hesaplanmasında kullanılmalıdır.
 - Bilgiler **input.txt**'de olduğu gibi boşluk ile ayrılmalıdır.
- Projede kullanılacak veri yapıları aşağıdaki gibidir:
 - *inis_pisti_kullanim_sirasi*: Havalimanına iniş yapacak uçakların iniş pistini kullanım sırasını içeren öncelikli kuyruktur. Uçak öncelik ve iniş saati bilgisi dikkate alınarak hazırlanacaktır.
 - *kalkis_pisti_kullanim_sirasi*: Havalimanından kalkış yapacak uçakların kalkış pistini kullanım sırasını içeren öncelikli kuyruktur. Uçak öncelik ve kalkış saati bilgisine göre hazırlanacaktır.

KISITLAR

- Proje C dili kullanılarak geliştirilecektir.
- Projede veri yapısı olarak öncelikli kuyruk kullanılmalıdır.

ÖDEV TESLİMİ

- Ödevin raporu **LaTeX** kullanılarak yazılmalıdır.
- Rapor IEEE formatında (önceki yıllarda verilen formatta) 4 sayfa olmalı, akış diyagramı veya yalancı kod içermeli, özet, giriş, yöntem, deneysel sonuçlar, sonuç ve kaynakça bölümünden oluşmalıdır.
- Dersin takibi projenin teslimi dâhil edestek.kocaeli.edu.tr sistemi üzerinden yapılacaktır. edestek.kocaeli.edu.tr sitesinde belirtilen tarihten sonra getirilen projeler kabul edilmeyecektir.
- <https://tr.overleaf.com/latex/templates/ieee-conference-template/grfzhncsfqn> adresinden IEEE formatlı örnek metin yapısını bulabilirsiniz.
- Proje ile ilgili sorular edestek.kocaeli.edu.tr sitesindeki **forum üzerinden** Arş. Gör. Kübra Erat veya Arş. Gör. Gamze Korkmaz Erdem'e sorulabilir.
- Sunum tarihleri daha sonra duyurulacaktır.
- Sunum sırasında algoritma, geliştirdiğiniz kodun çeşitli kısımlarının ne amaçla yazıldığı ve geliştirme ortamı hakkında sorular sorulabilir.
- Kullandığınız herhangi bir satır kodu açıklamanız istenebilir.

Sunum: Proje sunumu E-Destek üzerine yükleyeceğiniz projenizdeki kodlar indirilerek alınacaktır. Bu nedenle E- Destek üzerine yükleyeceğiniz projenin doğruluğundan emin olunuz.