**YZM 2124**

Veri Yapıları

**ÖDEV 1 RAPORU**

1. Giriş

Ödev bir kargo firmasına gelen müşterileri simüle ederek kargo sırası oluşturmak ve her bir kişiye 30 ile 300 saniye arasında bir işlem süresi vererek müşterileri sıralayarak toplam işlem süresini , ortalama işlem süresini ve müşterileri işlem süresine göre sıralamayı amaçlamaktadır.

Programa girdi olarak müşteri sayısının girilmesi gerekmektedir. Program başlatıldığında müşteri sayısı girilir daha sonra program sıfırdan başlayarak müşterilere id ve rastgele işlem süresi atar. Bu atamalar çıktı olarak konsola yazdırılır.

Kullanılan dosyalar ve metotlar

Ödev 1.1 de kullanılan stack.h liste veri yapısı kullanılarak tasarlandı. 1.2 de kullanılan stack.h dizi veri yapısı kullanılarak tasarlandı dizinin maksimum uzunluğunu veren *MAX\_SIZE* adında integer değişkenler tanımlandı ve sıralama yapmak için sorting(int length) metodu kullanıldı.

stack.h : Yığın oluşturmak için hazır kütüphane kullanmak yerine bir header dosyası oluşturuldu. İçinde yığın boyutunu tutan *stackBoyutu* adında bir integer tanımlandı. Bu yığın push(int data), pop(), empty(), top(), ust(), sorting(int length) metotlarından oluşmaktadır.

Metotlar

push(int data) : Yığına veri ekler. Ödev 1.1 ve 1.2 de aynı işlevi yapar.

pop(): Yığından veri çıkartır. Ödev 1.1 ve 1.2 de aynı işlevi yapar.

empty(): Yığının boş olup olmadığını kontrol eder ve Bool veri tipinde geri dönüş yapar. Yığından veri çıkartır. Ödev 1.1 ve 1.2 de aynı işlevi yapar.

top(): Yığına en son giren veriyi geri verir. Ödev 1.1 de kullanılır.

ust(): Yığına en son giren veriyi geri verir. Ödev 1.2 de kullanılır.

sorting(int length): Dizi elemanlarını küçükten büyüğe sıralar ve indislerine uygun şekilde yazdırır. Bu işlem için *temp* adında geçici bir dizi oluşturulur ve orijinal dizinin elemanları geçici diziye kopyalanır. Dizi uzunluğunu tutmak için *n* adında bir integer tanımlanır. Toplam süreyi hesaplamak için *topSure* adında bir integer tutulup kullanıcıların işlem süreleri hesaplanır. Ödev 1.2 de kullanılır.

main.cpp : Rastgele işlem süresi oluşturmak için kullanılan fonksiyon bu dosya içerisinde yer alır. Oluşturulan yığın ile kuyruk yapısı tasarlandı ve bu yapıları main metodunda bu dosya içerisinde çalıştırıldı. Bu dosya randomTime(int kisiSayisi, int kisiIslem[]), enQueue(int data), deQueue, main(), metotlarından oluşmaktadır.

Metotlar

randomTime(int kisiSayisi, int kisiIslem[]) : Girilen kişi sayısının ve main içerisinde oluşturulan kişileri tutan diziyi kullanarak kişi sayısı kadar 30 ile 300 arasında rastgele sayılar üretip, *random* adında bir integer da tutar ve gönderilen dizi içerisine yazmayı sağlar. srand(time(0)) ile her çalıştığında farklı sayılar üretir.

enQueue(int data) : Yığın oluştururken aynı zamanda bu yığından kuyruk yapısı tasarlamamıza yarar. Data parametresi kullanıcının işlem süresini gönderir. Gönderilen bilgi ile yığın oluşturulur.

deQueue() : Yığını kuyruğa eklemeye ve çıkarmaya yarayan metottur. Yığının boş olup olmadığını kontrol eder. Yığın boş değil ise yığın boşalana kadar yığının içinde bulunan verileri ikinci bir yığına atar. Bu şekilde yığını ters çevirip bir kuyruk oluşturmuş olur. Metodun sonunda ikinci kuyrukta bulunan verileri de silerek işlemi bitirir. Ödev 1.1 de silme sırasında silinen veriyi ekrana yazdırır.

Main() : İçinde kişi sayısını tutulduğu *kisiSayisi* adında integer veri tipi bulunur. kişiSayisi değişkenine kullanıcı konsoldan değer girer. Kişi sayısı uzunluğunda bir dizi oluşturulur ve rastgele verilen kişilerin işlem süresi dizinin içerisinde tutulur. Kişilerin bilgisi ve işlem süresi ekrana yazdırılır. Kişiler yığın kullanarak oluşturulan kuyruğa kaydedilir. Daha sonra ödev 1.1 de kuyruktan çıkarılır. Ödev 1.2 ise işlem süresine göre sıralanır ve kuyruktan çıkarılır. Toplam işlem süresi ve ortalama işlem süresi burada hesaplanır. Veri yapısı işlemleri de burada çağırılır.

1. Geliştirme ve Çalıştırma Ortamları

Ödevi Visual Studio kullanarak geliştirildi.

1. Veri Yapıları ve Değişkenler

Ödev geliştirilirken liste, dizi, yığın ve kuyruk veri yapılarını kullanıldı. Müşterileri ilk olarak dizi yapısı içerisinde tutulup daha sonra yığın veri yapısına eklendi. Ödev 1.1 de liste veri yapısı yığın veri yapısı oluşturmak için kullanıldı. Ödev 1.2 de yığın veri yapısı dizi kullanarak oluşturuldu. Yığın veri yapısı ise kuyruk veri yapısı ile birlikte kullanıldı.

1. Program Akışı

İlk olarak program müşteri sayısının girilmesini ister. Müşteri sayısı girildikten sonra müşteri sayısının uzunluğunda bir dizi oluşturur. Her bir kişiye rastgele işlem süreleri atar. Kişilerin işlemlerinin ne kadar süreceği ekrana yazdırılır. Kişiler yığın ile oluşturulan kuyruğa eklenir. Ödev 1.1 de giren müşteride başlayarak müşterileri kuyruktan çıkarır. Çıkarma işlemi gerçekleşirken konsola çıkan kişinin bilgileri yazdırılır. Ödev 1.2 de ise işlem sürelerine göre sıralama yaparak müşterileri kuyruktan çıkarır. Çıkarma işlemi gerçekleşirken konsola çıkan kişinin bilgileri yazdırılır. Bütün müşterilerin işlem sürelerini hesaplayıp toplam işlem süresini ve ortalama işlem süresini ekrana yazdırır.

1. Sonuç

Derste görmüş olduğum veri yapılarını daha iyi kavrama ve yakından deneyimleme fırsatı bulmuş oldum. Hazır veri yapısı kütüphanesi kullanmayarak kendi veri yapılarımı oluşturdum. Kendim bir sıralama algoritması geliştirip bunu kullandım. Bu sayede öğrendiklerim teoriğin yanında uygulamaya da geçmiş oldu. C++ dilinde kendime olan güvenimi arttırdı ve kendimi geliştirmemi sağladı. Bunun yanında visual studio kullanma becerilerime katkı sağladı ve ödev raporlarının nasıl yazılacağı hakkında bilgi sahibi oldum.