

EGE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
204 VERİ YAPILARI (3+1)
2008-2009 GÜZ YARIYILI
PROJE 2

Veriliş Tarihi : 21.10.2008
Teslim Tarihi : 09.11.2008 saat 12:00
Kontrol Tarihi : Asistanlar tarafından belirtilecek

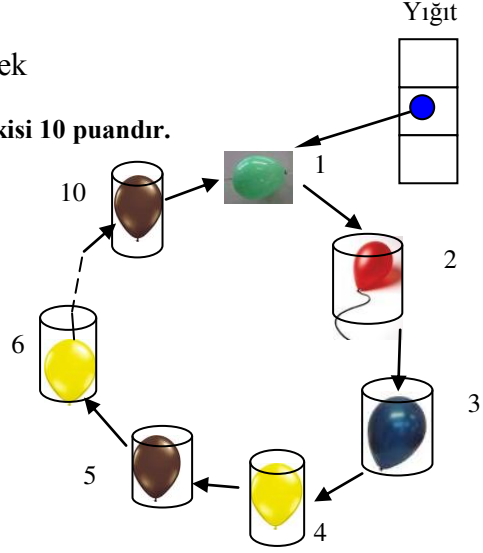
2. projenin arasınav notuna etkisi 10 puandır.

1) Balon problemi aşağıdaki gibi tanımlanmaktadır:

The following problem is called the Josephus problem:

m people stand in a circle, and an extra person goes round the circle eliminating every nth person. When the end of the circle is reached, we go round again, ignoring people who have already been eliminated, and still eliminating every nth person. The problem: given m and n, where should I stand in the circle so that I will be the last to be eliminated?

m tane balon, her bir yığıt elemanının en fazla 10 adet balondan oluşan bir bağlaçlı liste (dairesel olarak dizilmiş) alabildiği dikkate alınarak, yığıtın elemanlarına (katlarına) yerleştirilecektir.



Yığıtın en son eklenen elemanındaki ilk balondan başlanarak, aynı katta kalmak kaydıyla n. balon patlatılır (m ve n değerlerini kullanıcı belirtebilmelidir). Kalınan noktadan devam edilerek tekrar n tane balon sayılarak işlem tekrarlanır. İşleme, kalan son balonun rengi bulunana kadar devam edilir. Yığıtın ilgili katında balon kalmadığında işlem yığıtın bir alt katından devam ettirilir.

Java dilinde bu problemin çözümünü ve simülasyonunu yapan programı yazınız:

- Dairesel tek bağlaçlı listeler (random renklerde balonlardan oluşan) içeren yığıt (Java'nın hazır Stack sınıfı kullanılmalı) oluşturunuz ve her bir listeyi ekrana yazdırınız. Renkleri (Kırmızı, Yeşil, Mavi, ... gibi) "text" olarak temsil edebilirsiniz.
- Tüm balonlar bitene kadar işlemleri tekrarlayınız ve her turda, n değerini, kat numarasını, ilgili katta kalan balonların ve silinen renklerini ekrana yazdırınız.
- Kullandığınız bilgisayarın 1 saniyede ortalama kaç adet balon problemi çözebildiğini hesaplatınız. Kendinizin belirleyeceği üç farklı (m,n) kombinasyonu için sonuçları elde edip rapora ekleyiniz.

2) a) Bir bankaya sabahleyin gelmiş olan N kişinin, geliş sıralarına göre kuyruk oluşturduklarını düşünün. Her birinin işleminin 30 saniye ile 300 sn. arasında random bir zaman aldığını varsayın. Bunun için öncelikle, Java'da FIFO kuyruk yapısı (sınıf) tasarlayınız. Her bir banka müşterisine numara vererek ve her birine random birer işlem süresi atayarak N elemanlı bir kuyruk oluşturunuz. Her bir müşterinin işi biterek kuyruktan çıkarıldığında işlem tamamlanma süresini (işlem süresi dahil ne kadar süre kuyruқта kaldığını) liste halinde yazdırınız. Ayrıca, N müşteri için ortalama işlem tamamlanma süresini de hesaplayıp yazdırınız.

b) Aynı işlemi aynı değerleri kullanarak öncelikli kuyruk için tekrarlayınız. Kuyruğu, işlem süresi en kısa olan müşteri ilk (öncelikli) olarak hizmet alacak şekilde tasarlayınız (kuyruk, elemanları küçükten büyüğe sıralı tutacak). Her bir müşteri, işi biterek kuyruktan çıkarıldığında işlem tamamlanma süresini liste halinde yazdırınız. Ayrıca, N müşteri için ortalama işlem tamamlanma süresini de hesaplayıp yazdırınız. Bu yapıda, hangi müşterilerin FIFO kuyruğa göre, daha fazla beklediğini listeleyiniz (işlem süreleri ve sıra numaraları ile).

Projenin 1. ve 2. bölümleri için ayrı birer kısa rapor hazırlanmalıdır. Her iki programın açıklama satırları destekli kaynak kodları ile beraber, belirtilen tarihten önce, 2 veya 3 kişilik çalışma grupları tarafından ilgili oturum asistanına (duyuracağı e-posta adresine, belirteceği şekilde) teslim edilmelidir. Bu projede yazıcı çıktısı istenmemektedir.

Kapak : Proje numarası ve adı, öğrenci numaraları ve ad-soyadları, teslim tarihi bilgilerini içermelidir.

Kısa Rapor Formatı

1. Gerçekleştirilen Platform ve Dil (Java, Eclipse?) ve Sürüm Adı
2. Problemin kısa tanımı (ödevden)
3. Kullanılan veri yapılarının, sınıfların ve metotlarının kısa açıklamaları
4. Yazılım Geliştirme İçin Harcanan Süreler (kişi ve saat bazında)
5. Elde edilen örnek sonuçlar

Raporun kapak dahil 10 sayfayı geçmemesi önerilir; sayfaları numaralandırılmalıdır.