

Number :
Name :

B

22.11.2011

EGE UNIVERSITY
FACULTY of ENGINEERING
COMPUTER ENGINEERING DEPARTMENT
204 DATA STRUCTURES (3+1)
FALL 2011
MIDTERM EXAM (120 minutes)

Exam : 85 Points (Project 1 : 5 and Project 2 : 10 pts)

There is no penalty for minor Java syntax errors, or slightly misnamed methods, if it's clear you have the right idea. Explaining your answer may help us give you partial credit if it's wrong.

1) (20 pts)

- (15) a) Write a Java **method** (floodFill) filling (or recoloring) a particular bounded area with color using recursion (Figure 1–next page). [Sınırlandırılmış bir iç (beyaz) bölgeyi renklendirme işlemini özyineli bir şekilde yapan **Java metodunu** yazınız].
- (5) b) Explain method (algorithm) you developed in your own words. [Yönteminizi anlatınız]

Use the 4-direction flood fill branching in 4 directions from each pixel.

int getPixel(int x, int y) : *Read* the *color* of the *pixel* at the given coordinates

putPixel(int x, int y, int color) : Sets *pixel color* at coordinates (x,y).

Ek Açıklama: Ekranda, farenin tıklandığı noktadan itibaren 4'lü komşuluğa göre (4 ana yönden) bağlı beyaz bölgeyi (veya siyah piksellerle çevrili bir alanın tümünü), verilen bir renkle (kırmızı) doldurması istenmektedir. Çapraz hareketlere izin verilmemektedir.

- Herhangi bir x,y konumundaki pikselin renginin getPixel(x,y) metodu ile elde edilebileceğini ve x,y konumundaki pikselin renginin setPixel(x,y,Color) metodu ile Color içerisinde belirtilen renge dönüştürülebileceğini varsayabilirsiniz.
- Renk karşılaştırmaları için Kırmızı, Beyaz ve Siyah renk sabitlerinin önceden tanımlandığını varsayarak doğrudan kullanabilirsiniz.
- Kapalı bir alan içerisine tıklandığını ve metodunuza ilk x,y konumunun gönderildiğini varsayabilirsiniz (Ekran sınır kontrolünü yaptırmanıza gerek yoktur).

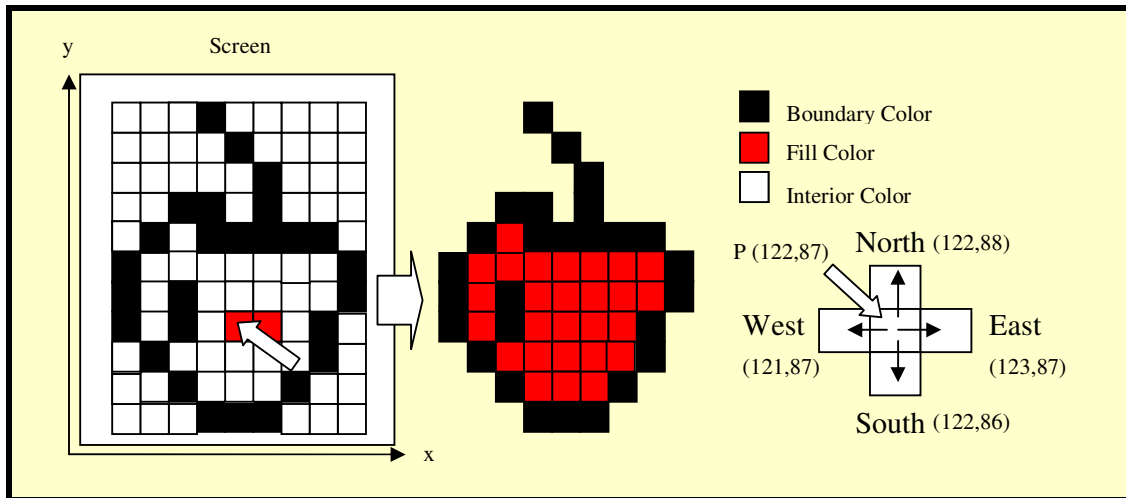


Figure 1: Area Filling Example (inside a Cherry)

2) (20 pts)

Write a Java **class** (also **constructor**, **enqueue** and **dequeue** methods) as a data structure simulating queues at a supermarket.

Enqueue: Customers must choose the first suitable checkout lane. Suitable means not exceeding 3 customers (max. 3). If all checkouts exceed this limit, additional checkout should be opened. You can prefer and use any of the built-in dynamic data structures such as Vector, ArrayList, Queue or LinkedList for holding checkouts and queues at checkouts.

Dequeue: *removes* the *customer* at the front of the queue at random checkout and calculates **average waiting time** (for all customers) after a customer has been served. Idle checkouts should be closed.

(**Waiting time** of a customer is the **number of customers** in the queue while joining a queue) Solutions including array instead of dynamic data structure will incur %50 penalty. It is not needed to write extra method for testing.

[24 saat çalışan bir süpermarketteki kasalarda oluşan kuyrukların simülasyonunun yapılması için basit bir veri yapısı (sınıf) tasarlayınız. Altyapıda (Bileşik Listeyi tutmak için) Queue, “ArrayList”, “Vector”, Bağlılı Liste veri yapılarından istediklerinizi seçebilirsiniz. Hiç kasanın açık olmadığı durumdan başlanarak (yapılandırıcı), 3’ten az sayıdaki müşterinin beklediği ilk kasadaki kuyruğun sonuna müşteriye ekleyen, açık tüm kasaların 3 elemanlı olması durumunda yeni bir kasa açarak müşteriye kuyruğuna ekleyen ekle (enqueue) metodunu yazınız. Random bir kasadaki müşterinin işlemini tamamlayan ve (o ana kadar olan) ortalama işlem tamamlanma süresini (kişi sayısı cinsinden) hesaplayan sil (dequeue) metodunu yazarak, kasada bekleyen müşteri kalmadıysa ilgili kasayı kapatmasını da sağlayınız (ArrayList mantığı). Elemanları Dizi’de tutan, 10 puan üzerinden değerlendirilir. Yapıyı kullanan ayrı bir main metodu yazmanıza gerek yoktur!]

3) (15)

a) Write an **efficient** Java method (or algorithm) to check (detect) whether a given **binary tree** is **full**. [Verilen bir ikili ağacın Full Binary Tree olup olmadığını bulan etkin Java metodunu yazınız.] (10)

b) Explain the method (or algorithm) you proposed. [Önerdiğiniz algoritmayı veya yönteminizi açıklayınız.] (5)

Note: Don’t write Tree, TreeNode classes and other methods. Assume that Tree, TreeNode classes and other methods in course notes are ready. Your method is allowed to call another method you write.

Not : Ağaç, Düğüm sınıfları ile diğer metotlarını yazmayınız. Ders notlarındaki hazır Tree ve TreeNode sınıfları ile ilgili diğer metotların hazır geldiğini varsayabilirsiniz. Metodunuz, yazacağınız başka bir metodu çağırabilir.

4) MULTIPLE CHOICE QUESTIONS (3 pts each)

Sınavın Çoktan Seçmeli Test Bölümü

Arka sayfadaki test sınavı, her biri 3'er puan olan 10 sorudan oluşmaktadır ve toplam 30 puan değerindedir. Yanlışlar doğruları götürmemektedir.