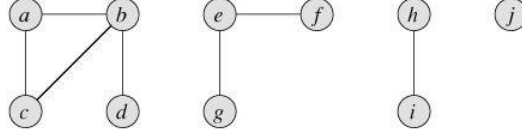


2012-2013 Bahar Yarıyılı
Veri Yapıları ve Algoritmalar 2. Ödevi

Konu : Grafın Bağlı Bileşenlerinin (Connected Component) Bulunması

Problem: Bir graf kendi içinde bağlı olan birden fazla alt graflardan oluşabilir. Bu alt graflar bağlı bileşen olarak isimlendirilir. Bu çalışmada, bir grafın **kaç bağlı bileşenden oluştuğu** bulunacak ve **her bileşendeki düğüm sayısı ve bu düğümlerin numaraları** verilecektir.

Örnek: Aşağıdaki graf 4 tane bağlı bileşenden yani 4 tane alt graftan oluşmaktadır.



Giriş :

1. Düğüm sayısı : **N**
2. Düğümler arası bağlantıları gösteren kenar sayısı : **M**
3. M adet bağlantı için, hangi düğümler arasında bağlantı olduğu:
Örneğin **1 4** sırası ile verilen bir giriş **1. düğümün 4. düğüme** bağlı olduğunu gösterir.

Çıkış:

1. **Kaç tane** bağlı bileşen olduğu
2. Her bileşendeki **düğüm sayısı**
3. Her bileşende **hangi düğümlerin bulunduğu**

Teslim Edilecekler: Algoritmanızın yazı ile açıklamasını, akış diagramını ve **C dilinde yazılmış** programını içeren **CD'yi 9 Nisan 2013 Çarşamba günü** yapılacak laboratuvarıda teslim ediniz.

Değerlendirme: Ödeviniz aşağıdaki gibi değerlendirilecektir :

Algoritma Tasarımı ve Programın Çalışması: (%80)

1. Ödev, istenilen işlerin tamamını yerine getirmelidir.
2. İşlemler modüler yapıda, fonksiyonlara bölünerek yapılmalıdır.
3. Gereksiz kontrollerden ve işlemlerden arınmış bir tasarım yapılmalıdır.
4. Program hatasız çalışmalıdır.
5. Programın çalışması sırasında, konuyu bilmeyen kişilerin rahatlıkla anlayabilmesi için, giriş ve çıkışlarda mesajlarla bilgi verilmelidir.

Rapor Dokümantasyonu: (%20)

1. Ödev A4 kağıda, yazıcıdan çıkış alınarak verilmelidir.
2. Raporun kapak sayfasında, dersin adı, öğrencinin ad, soyad ve numarası, ödev konusu bilgileri yer almalıdır.
3. Kod yazımına başlamadan önce kısaca problem tariflenmeli, yapılan işlemler kısaca anlatılmalıdır.
4. Kaynak kodda değişken deklarasyonu yapılırken her değişken tek satırda tanımlanmalı, tanımın yanına değişkenin ne için kullanılacağı açıklama olarak yazılmalıdır.
5. Değişken ve fonksiyon (veya metod) isimleri anlamlı olmalıdır.
6. Her fonksiyonun (veya metodun) yaptığı iş, parametreleri ve dönüş değeri açıklanmalıdır.
7. Gerekli yerlerde açıklama satırları ile kodda yapılan işlemler açıklanmalıdır.
8. Kaynak kodun formatı düzgün olmalıdır.