EGE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
204 VERİ YAPILARI (3+1)
2008-2009 GÜZ YARIYILI
PROJE 4 : ÇİZGELER (GRAPH) ve ARAŞTIRMA

Veriliş Tarihi : 16.11.2008
Teslim Tarihi : 05.01.2009

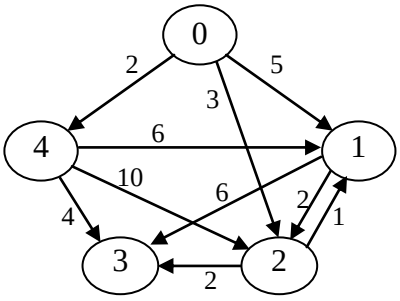
I - EN KISA YOL - SHORTEST PATH (Dijkstra's Algorithm) ve DİĞER ÇİZGE ALGORİTMALARI

Aşağıdaki işlemleri gerçekleştiren Java / C / C# / C++ programını yazınız.

1) Verilen "iller.txt" ve "maliyet.txt" dosyalarındaki çizge bilgilerinin bellekte oluşturulan veri yapılarına aktarılması (maliyet bilgileri için "weight" matrisi kullanınız).
2) Bellekteki maliyet matrisindeki bilgilerin adları (il adı) ile beraber ekrana listelenmesi. Klavyeden numarası verilen köşenin gelen ve giden kenarlarının ekrana listelenmesi (Verilen köşe numarası ve adı, gelen kenar sayısı, giden kenar sayısı ve her bir bağlı kenar için kenar değeri, bağlı olduğu köşe numarası ve adı ekrana yazılacak).
3) Klavyeden verilen bir köşe çifti arasındaki en kısa yolun bulunması.
4) Klavyeden verilen köşe numarasından başlanarak çizgenin **önce genişliğine (Breadth-First) dolaşılması.**
5) Çizgenin **en küçük kapsayan ağacının (minimum spanning tree)** bulunması (1 yöntemle).

Seçimlik : Matris elemanlarının, oluşturulan nxn'lik metin kutuları matrisinden girilebilmesi gibi GUI (Graphical User Interface) düzenlemeleri ve çizgenin grafiksel çizimi. Verilen bir köşe numarasından tüm diğer köşelere en kısa yolların bulunması. Çizgenin önce derinliğine (Depth-First) dolaşılması. Nearest Neighbor, Maximal Flow gibi diğer çizge algoritmaları. Verilen bir köşe (il) adının çizgede aratılması, işlemlerin ada göre de yapılabilmesi...

Iller.txt	**Maliyet.txt**				
5					
Ankara	∞	5	3	∞	2
İstanbul	∞	∞	2	6	∞
İzmir	∞	1	∞	2	∞
Eskişehir	∞	∞	∞	∞	∞
Kayseri	∞	6	10	4	∞



"iller.txt" dosyasında toplam köşe sayısı (illerin sayısı) ve köşe adları (illerin isimleri) tutulmaktadır. İlk satırda sadece 1 ile 16 arasında bir tamsayı "n" yani çizgedeki toplam köşe sayısı bulunmaktadır. Sonraki satırlarda sırası ile, 0. köşenin (ilin) adı, 1. köşenin adı, ... , (n-1). köşenin adı yer almaktadır.
"**maliyet.txt"** dosyası, yönlü çizgenin köşeleri arasındaki kenarların maliyet bilgilerini (değerlerini) içermektedir. "iller.txt" dosyasında belirtilen sayı kadar satırdan oluşmaktadır. Satır numarası kaynak köşe numarasını belirtmektedir. m. satırın n. elemanı, m. köşe ile n. köşe arasındaki kenarın değerini (maliyetini) göstermektedir. Maliyetler herhangi bir pozitif reel sayı olabilmektedir (Kelimeler arasında birer boşluk olduğu gibi kolaylaştırıcı varsayımlar yapılabilir.)

II - ARAŞTIRMA + ARAŞTIRMA KODU BÖLÜMÜ

Aşağıdaki Konulardan Sadece Birisini Seçerek Veri Yapıları ve Algoritmalar Yönü Ağırlıklı Olacak Şekilde Anlatınız. Konu ile ilgili hazır kodlardaki veri yapıları kullanımını inceleyerek, bir program yazınız (herhangi bir programlama dilini tercih edebilirsiniz):

1) **BTREE**
2) **HUFFMAN ENCODING**
3) **COMPRESSION ALGORITHMS (Genel)**
4) **LOSSY COMPRESSION (Kayıplı sıkıştırma ayrıntısı, image compression gibi)**
5) **LOSSLESS COMPRESSION (Kayıpsız sıkıştırma ayrıntısı)**
6) **MINIMUM SPANNING TREE**
7) **DYNAMIC PROGRAMMING**
8) **COMPUTATIONAL GEOMETRY**
9) **PROBABILISTIC ANALYSIS and RANDOMIZED ALGORITHMS**
10) **ARTIFICIAL NEURAL NETWORKS**
11) **GENETIC ALGORITHMS**
12) **TRIE**
13) **Diğer [Veri Yapıları ile ilgili araştırmayı düşündüğünüz diğer konuları danışınız]**

Örnekler: Araştırma konusu olarak, Yapay Sinir Ağları (YSA) seçildiyse, bir Algılayıcı (Perceptron) veya Çok Katmanlı Algılayıcı (MLP) kodu yazılabilir. Genetik Algoritmalar (GA) seçildiyse, 8-Vezir Problemini veya Gezgin Satıcı Problemini bu yöntemle çözen bir program yazılabilir. Sıkıştırma algoritmaları için, literatürde yer alan basit bir sıkıştırma kodu yazılabilir.

Not:

1) Sekiz Vezir Problemi: Bir satranç tahtasına 8 tane vezirin birbirini alamayacak şekilde yerleştirilmesini içerir.
2) "TSP (Traveling Salesman Problem) : Given a number of cities and the costs of traveling from any city to any other city, what is the cheapest round-trip route that visits each city exactly once and then returns to the starting city? An equivalent formulation in terms of graph theory is: Given a complete weighted graph (where the vertices would represent the cities, the edges would represent the roads, and the weights would be the cost or distance of that road), find a (Hamiltonian Cycle) with the least weight. (Wikipedia)

Araştırmada Yararlanabilecek Kaynaklardan Bazıları:

http://en.wikipedia.org/wiki/List_of_data_structures
http://en.wikipedia.org/wiki/List_of_algorithms
http://en.wikipedia.org/wiki/List_of_terms_relating_to_algorithms_and_data_structures
İnternet (google, …)
Dersin sayfasındaki kaynaklar.

Projenin Değerlendirmesi: Çizge Algoritmaları:5, Rapor+Araştırma Metni:5, Araştırma Kodu:5 puan.

Rapor Formatı

İlk sayfada, Proje numarası ve adı, öğrenci numaraları ve ad-soyadları, teslim tarihi bilgilerini içermelidir (Ayrıca kapak sayfası yapmaya gerek yoktur).

1. **Programcı Kataloğu (En kısa yol)**
    - 1.1 Gerçekleştirilen Platform ve Dil (Java, Eclipse?) ve Sürüm Adı
    - 1.2 Problemin kısa tanımı
    - 1.3 Kullanılan sınıfların ve metotlarının kısa açıklamaları
    - 1.4 Kullanılan veri yapıları ve kısa açıklamaları
    - 1.5 Kullanılan dosyaların özellikleri ve kısa açıklamaları
    - 1.6 Yazılım Geliştirme İçin Harcanan Süreler (kişi ve saat bazında)
2. **Kullanıcı Kataloğu (En kısa yol)**
    - 2.1 Programın İşletimi ve Ekran Görüntüsü (en az 1 tane)
    - 2.2 Kullanıcı Kılavuzu
    - 2.3 Programın Kısıtlamaları
3. **Araştırma ve Araştırmanın Kaynak Kodu**
4. **Kaynaklar**

Raporun 10 sayfayı geçmemesi önerilir (sayfalar numaralandırılmalıdır). Raporun Çıktısının, Dersi Veren Öğretim Üyesine son teslim tarihi 05.01.2009 saat 16:30'dur. Ayrıca Word Belgesi, kaynak kod ve veri dosyaları sıkıştırılarak, aybars.ugur@ege.edu.tr e-posta adresine (Önceki Projelerde olduğu gibi, No ve Adsoyadlar belirtilerek) gönderilmelidir. Programlar makine başında kontrol edilmeyecek, koddan incelenecektir.