

**2018-2019 Güz Yarıyılı**  
**Algoritma Analizi Dönem Projesi**

**Konu:** Graf İşlemleri

**Problem:** Harita üzerinde bir noktadan diğerine en kısa yoldan gitme

Pekçok GPS Navigasyon sistemi bir adresten istenilen bir adrese gitmek için BFS(Breadth First Search) algoritması kullanmaktadır. Bu ödevde graf üzerinde bu işlemi yapan bir algoritma tasarlanacaktır.

**Yapılacak İşlemler :** Verilen bir **yönlü graf** üzerinde, verilen bir başlangıç düğümünden istenilen düğüme en kısa yoldan ulaşmanız istenmektedir. Graf yapısını derste gösterildiği şekilde **adjacency list** kullanarak oluşturunuz. **Grafın hazırlanması, okunması ve BFS algoritmasının temel işlemleri için derste gösterilen algoritmayı kullanınız.** Bir düğümden diğer düğüme en kısa yolu bulmak için gerekli algoritma tasarımını kendiniz yapınız.

İşlem adımları :

1. N düğümlü, M bağlantılı grafın bağlantılarını ve bağlantıların değerlerini okuyun.
2. **Başlangıç** ve **Varış** düğümlerini okuyun.
3. **Başlangıç** düğümü ile **Varış** düğümü arasında birden fazla yol olabilir. Bütün yolları tek tek **BFS** yöntemi ile bulunuz.
4. Bulduğunuz yollardan **en kısıasına ait düğümler ve düğümler arası bağlantı değerleri bilgilerini** ekrana yazdırınız.
5. Belirtilen düğümler arasında bağlantı yoksa gidişin mümkün olmadığını belirten bir mesaj verilmelidir.
6. Haritada birbirine doğrudan bağlı olan düğümlerin listesini ekrana yazdırınız.
7. En fazla komşusu olan düğümü ekrana yazdırınız.
8. Haritada birbirine en uzak iki noktayı ve uzaklıklarını ekrana yazdırınız.

Dikkat Edilmesi Gereken Noktalar:

1. A noktasından B noktasına gidiş, B noktasından A noktasına gidiş ile aynı yol olmayabilir.

**Teslim Edilecekler:** Aşağıda verilen **bütün bilgileri içeren** tek bir doküman hazırlayınız. Raporun kapak sayfasına, dersin adı, öğrencinin ad, soyad ve numarası, ödev konusunu yazınız.

1. Yaptığınız çalışmayı **yöntem, uygulama ve sonuç** bölümlerinden oluşan bir raporda anlatınız.
  - **Yöntem Bölümü :** Problemin tanımını verip, gerçekleştirdiğiniz çözümü kısaca anlatınız.
  - **Uygulama Bölümü :** Hazırlayacağınız küçük bir graf üzerinde tasarladığınız algoritmanın analizini yapınız. Farklı örnekler için ekran çıktıları alınız (en az 5 farklı örnek)
  - **Sonuç Bölümü :** Algoritmanızın karmaşıklığını hesaplayınız.
2. Algoritmanızın **kodunu C dilinde** yazınız.

**Teslim İşlemleri:**

Ödevler 2.Final Haftasında daha sonra belirlenecek bir tarihte laboratuvarında gösterilecektir. Geç teslimler değerlendirilmeyecektir. Ödev teslim kuralları ve sunum planı için Arş. Grv. Ahmet ELBİR'in sayfasını takip ediniz.

- Teslim edilecek rapor ve program kodu ismi **ÖğrenciNumarası\_Ad\_Soyad** olan **klasör** içinde olmalıdır.
- Bu klasör içinde **rapor** ve **kod** klasörü olmalıdır.
- Rapor, **ÖğrenciNumarası\_Rapor.pdf** şeklinde adlandırılmalıdır.
- Kod klasöründe kaynak kod yer almalıdır.

**Laboratuvar Sunumu:** Programınızın çalışmasını laboratuvar esnasında size verilecek olan örnek bir graf için göstermeniz istenecektir.

**Değerlendirme:** Ödeviniz aşağıdaki gibi değerlendirilecektir:

***Algoritma Tasarımı ve Programın Çalışması: (%60)***

1. Ödev, istenilen işlerin tamamını yerine getirmelidir.
2. Gereksiz kontrollerden ve işlemlerden arınmış bir tasarım yapılmalıdır.
3. Programda gerekli alt modüller belirlenerek her modül ayrı fonksiyon olarak yazılmalıdır.
4. Program hatasız çalışmalıdır.
5. Programın çalışması sırasında, konuyu bilmeyen kişilerin rahatlıkla anlayabilmesi için, giriş ve çıkışlarda mesajlarla bilgi verilmelidir.

***Rapor Dokümantasyonu: (%40)***

1. Raporun ilk sayfasında, dersin adı, öğrencinin ad, soyad ve numarası, ödev konusu bilgileri yer almalıdır.
2. Kaynak kodda değişken deklarasyonu yapılırken her değişken tek satırda tanımlanmalı, tanımın yanına değişkenin ne için kullanılacağı açıklama olarak yazılmalıdır.
3. Değişken ve fonksiyon(veya metod) isimleri anlamlı olmalıdır.
4. Her fonksiyonun (veya metodun) yaptığı iş, parametreleri ve dönüş değeri açıklanmalıdır.
5. Gerekli yerlerde açıklama satırları ile kodda yapılan işlemler açıklanmalıdır.
6. Gereksiz kod tekrarı olmamalıdır.
7. Kaynak kodun formatı düzgün olmalıdır.