Prof. Dr. Mine Elif KARSLIGİL BLM2512 Veri Yapıları Ve Algoritmalar

**Son Teslim tarihi:** **21.05.2024 23:59**

Proje

Oğuzhan Topaloğlu

Ç19052025 – Grup 1

*Bilgisayar Mühendisliği Bölümü,*

*Elektrik-Elektronik Fakültesi,*

*Yıldız Teknik Üniversitesi*

****

Istanbul, 2024

**Video Linki:** [**https://www.youtube.com/watch?v=eyypf7IcPII**](https://www.youtube.com/watch?v=eyypf7IcPII)

|  |
| --- |
| **İÇERİK** |
| 1. Problemin Tanımı |
| 2. Problemin Çözümü |
| 3. Karşılaşılan Sorunlar |
| 4. Ekran Görüntüleri |
|  |
|  |
|  |
|  |
|  |

**1. Problemin Tanımı**

Bu projede, belirli bir ağırlığa (weight) sahip çeşitli kenarları (edge) olan bir graf üzerindeki tüm olası çokgenlerin (polygon) belirlenmesi gerekmektedir. Çokgenler, grafın düğümlerini (node’larını) birleştiren kenarlarla oluşturulur ve her biri belirli bir ağırlığa sahiptir. Bu ağırlıklar, her çokgenin toplam uzunluğunu belirlemektedir. Bu problem, verilen bir graf üzerindeki tüm olası çokgenleri bulmayı ve bunları belirli bir düzende organize etmeyi içerir.

**2. Problemin Çözümü**

Problemin çözümü öncelikle bir graf oluşturarak başlar. Bu graf, bir txt dosyasından okunan verilere dayanır ve her düğüm, bir komşuluk/kenar listesi ile temsil edilir. Her kenar, bir hedef düğüm ve bir ağırlığa sahiptir.

Daha sonra, derinlik öncelikli arama (DFS) algoritması kullanılarak graf üzerindeki tüm olası çokgenler belirlenir. DFS, her düğümden başlar ve ziyaret edilen düğümleri bir listede saklar. Eğer DFS, başlangıç düğümüne geri döner ve listede en az üç düğüm varsa bir çokgen bulunmuştur. Bu çokgen bulunduktan sonra çokgen linkli listesine eklenir.

Bir sonraki adımda, çokgenler alfabetik sıraya göre düzenlenir ve aynı çokgenlerin farklı sıralamaları (örneğin, ABCD ve ADCB aynı çokgendir) listeden çıkarılır.

Son olarak, her bir çokgenin detayları ve toplam çokgen sayısı yazdırılır. Bu detaylar, çokgenin köşe sayısını, içerdiği kenarları ve toplam uzunluğunu içerir.

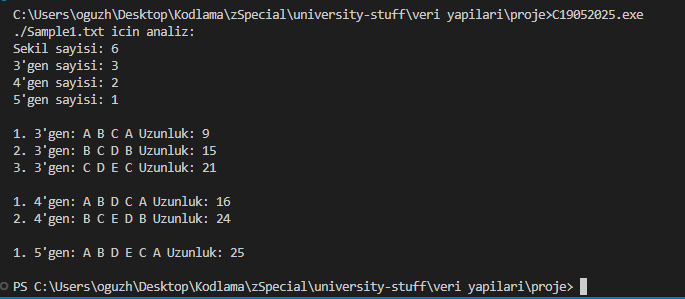
**3. Karşılaşılan Sorunlar**

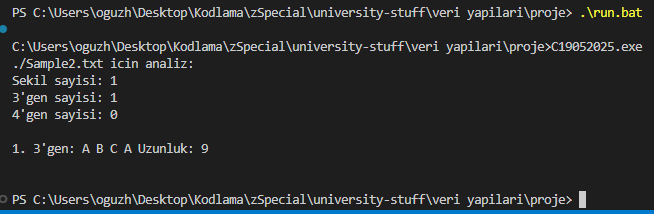
En çok sorun void organizePolygons(Polygon\*) fonksiyonunda ortaya çıkmıştır. Fonksiyon içinde alfabetik olarak en küçük sıralamayı bulmak, her yolu (path’i) alfabetik şekilde sıralamak/döndürmek ve kimi yolların tam tersinin varlığını tespit etmede çok hata yapılmıştır. String işlemleri ile karşılaştırma yapılırken sık sık null termination kavramını ve en sonlarına ‘\0’ eklemeyi unuttum.

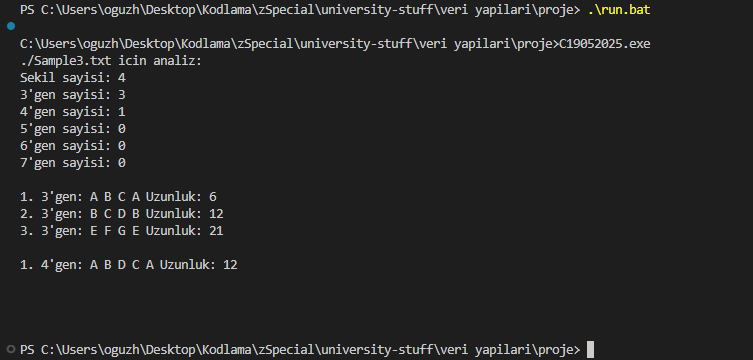
Ayrıca void printResults(Graph\*, Polygon\*) fonksiyonunda konsola her bulunan çokgen yazdırılırken indekslemelerde bir sürü hata yaptım.

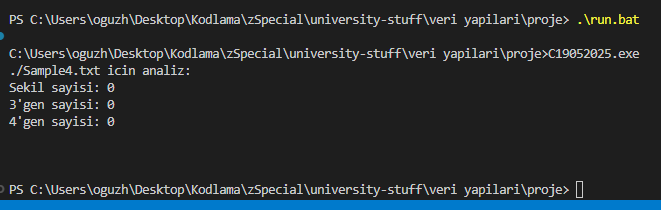
DFS algoritmasında da uzunluğu 2 olan sonuçları ayıklamakta biraz sıkıntı çektim.

**4. Ekran Görüntüleri**

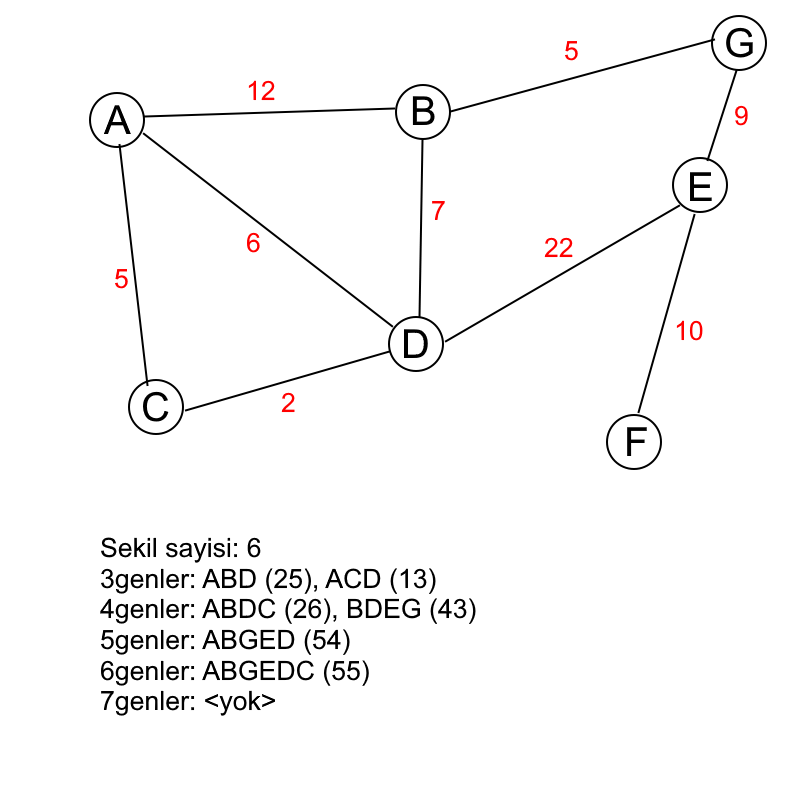
****

****

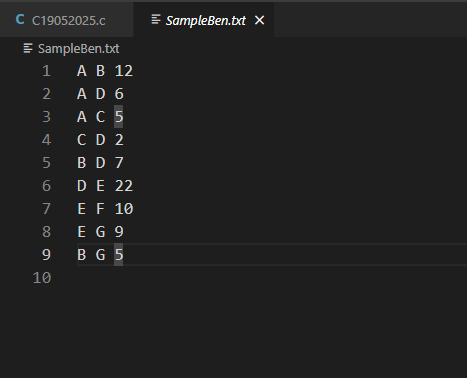
****

****

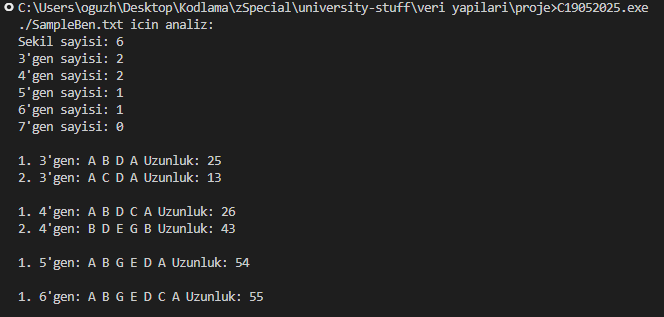
Kendi örneğim ([paint.net](https://www.getpaint.net/) isimli uygulamada oluşturdum):

****

Kendi örneğimin dosya şekli:

****

Kendi örneğimin çıktısı:

****