

**T.C. İSTANBUL OKAN ÜNİVERSİTESİ**

**MESLEK YÜKSEK OKULU**

**BÖLÜM ADI: ELEKTRONİK VE OTOMASYON BÖLÜMÜ**

**PROGRAM ADI: MEKATRONİK**

**DERS ADI: YAPAY ZEKA**

**PROJE ADI: KRUSKAL ALGORİTMASI VE MANTIĞI**

**HAZIRLAYAN VE SUNAN**

**AD SOYAD: SİBEL MİRAÇ ÖZCAN**

**OKUL NUMARASI: 18MY24033**

**DERSİN ÖĞRETİM ÜYESİ: NİLGÜN İNCEREİS**

**DANIŞMAN: TARIK ASLAN**

İçindekiler

[GİRİŞ 3](#_Toc39378461)

[1. GENEL BİLGİLER 3](#_Toc39378462)

[1.1. Yapay Zeka Nedir? 3](#_Toc39378463)

[1.2. Yapay Zekada Minimum Yolun Bulunması Problemi Nedir? 3](#_Toc39378464)

[2. PROJE BİLGİLERİ 6](#_Toc39378465)

[2.1. Projemin Konusu 6](#_Toc39378466)

[2.2. Projemin Adı 6](#_Toc39378467)

[2.3. Projemin Programlama Dili 6](#_Toc39378468)

[2.4. Projemi Programladığım Platform 6](#_Toc39378469)

[2.5. Projemin Minimum Yolun Bulunması Uygulaması Tasarımı 6](#_Toc39378470)

[2.6. Projemin Algoritması 6](#_Toc39378471)

[2.7. Projemin Kodlaması 7](#_Toc39378472)

[2.8. Projemin Ürettiği Ekran Çıktıları 9](#_Toc39378473)

[2.9. Projemi Nasıl Yaptım? 10](#_Toc39378474)

[3. ÖZ GEÇMİŞ 11](#_Toc39378475)

[4. KAYNAKÇA 11](#_Toc39378476)

[5. EK 11](#_Toc39378477)

[5.1. Proje Kodları Ana Başlıkları 11](#_Toc39378478)

[6. PROJEMİN YAYINLANMASI 12](#_Toc39378479)

[6.1. GitHub Linki 12](#_Toc39378480)

# GİRİŞ

## GENEL BİLGİLER

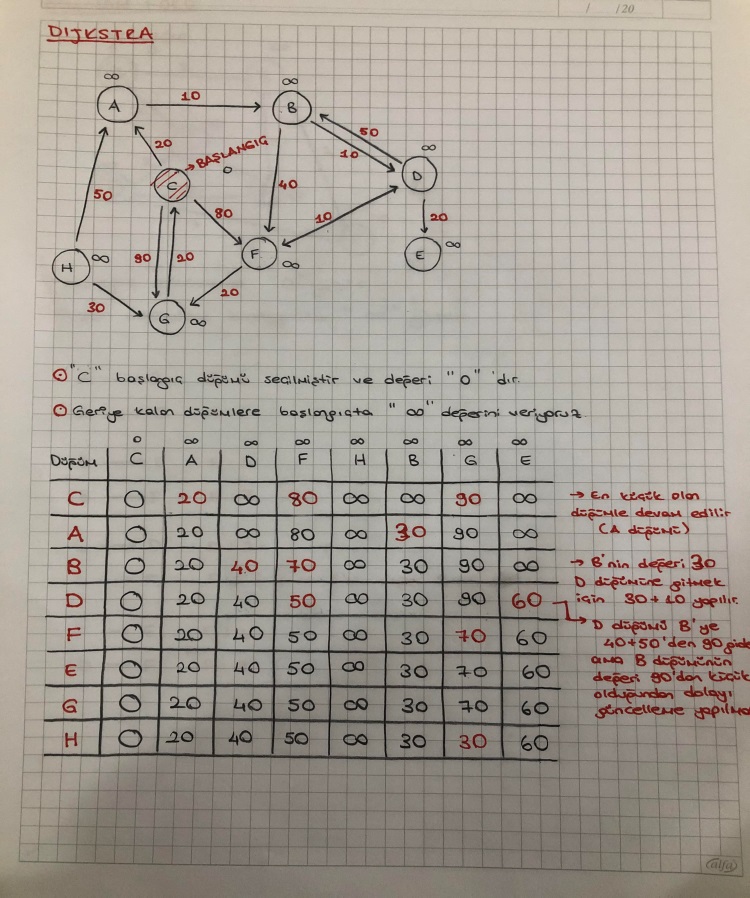
### Yapay Zeka Nedir?

* Yapay Zeka; bilgisayar sistemine bağlanmış olan bir robotun canlılara benzer şekilde bazı eylemleri yerine getirme kabiliyetine sahip olan bir icattır. 1940’lı yıllarda Turing Machine’lere kadar giden bu icat biz insanların düşünme şekillerini analiz ederek oluşturulmuş ve geliştirilmeye devam edilmiştir.

### Yapay Zekada Minimum Yolun Bulunması Problemi Nedir?

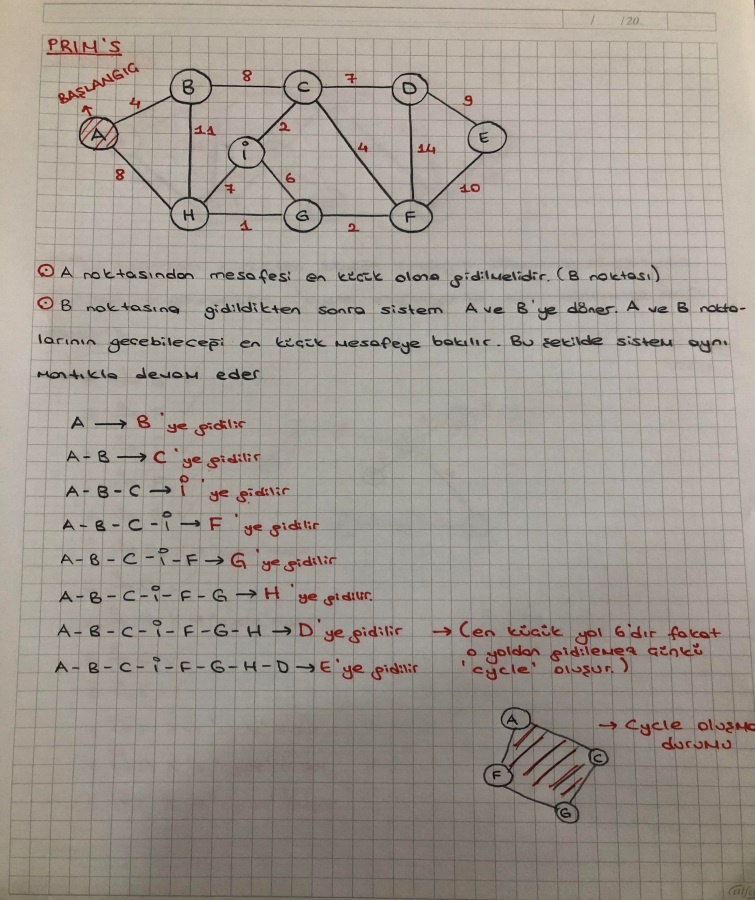
#### Dijkstra Algoritması

* Dijkstra Algoritmasında amacımız bulunduğumuz noktadan en az maliyet ve en kısa yoldan çözümü bulma amacı taşımaktır.
* Başlangıçta başlamak istediğimiz düğüme ‘0’ maliyeti veririz ve diğer düğümlerin hepsine ‘∞’ yazarız.



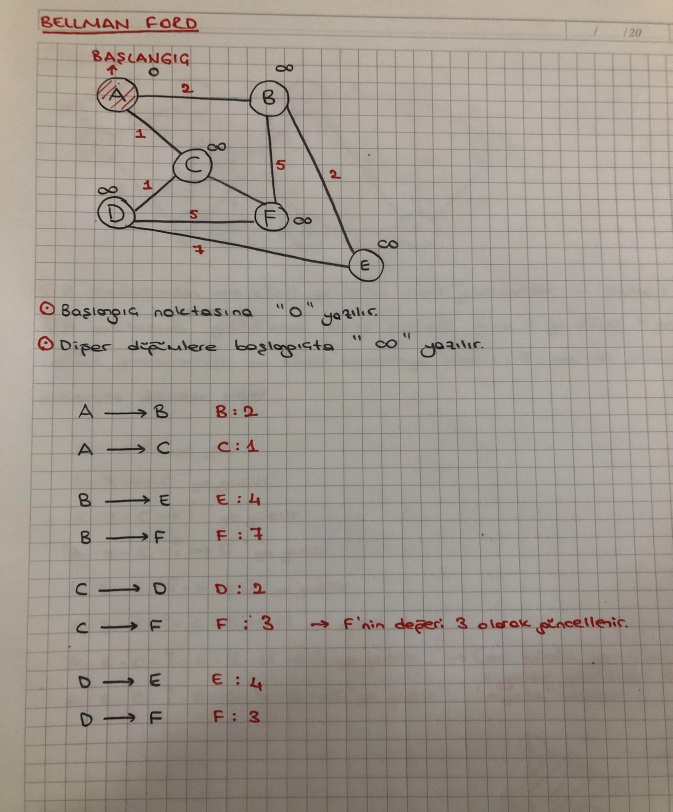
#### Prim’s Algoritması

* Prim’s Algoritması Minimum Spanning Tree’ye dayanmaktadır.
* Graf yapısında ağaç oluşturur ve var olan tüm düğümlere en kısa yoldan erişmeyi sağlamaktadır.
* Dikkat edilmesi gereken tek nokta ise ‘Cycle’ oluşmasını önlemektir. Yani döngü oluşmaması gerekmektedir.



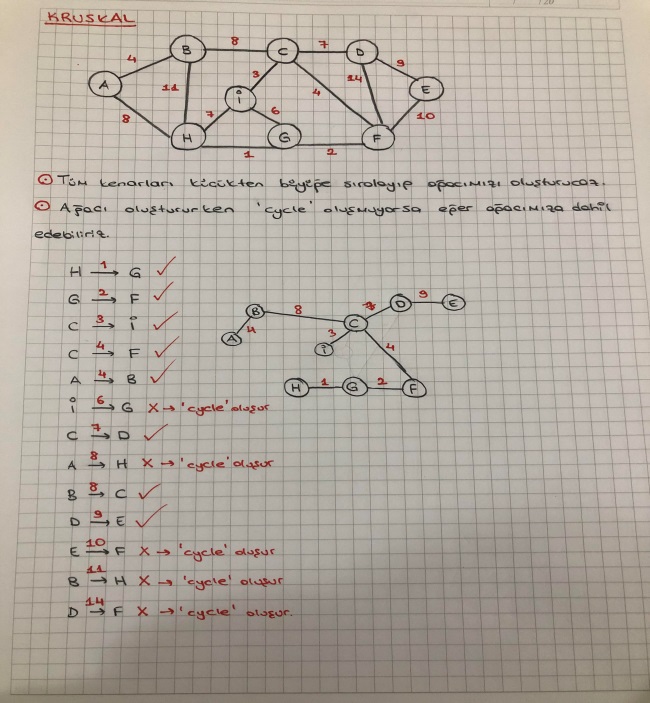
#### Bellman Ford Algoritması

* Dijkstra Algoritmasıyla fazlasıyla benzerlik gösteren bir algoritmadır.
* Başlangıç düğümünden itibaren graf yapısı içindeki tüm düğümleri dolaşarak en kısa yolu hesaplamaktadır.
* Dijkstradan daha kötü performansa sahip dersek yanılmış olmayız.
* Başlangıç düğümüne ‘0’ verilir ve diğer düğümler ‘∞’ olarak güncellenir.



#### Kruskal Algoritması

* Prim’s Algoritmasıyla benzerlik gösterir ve bir nevi Prim’s Algoritmasının kardeşi olarak da düşünebiliriz.
* Öncelikle tüm kenarlar küçükten büyüğe sıralanır, daha sonra ağaç oluşturulur fakat ağacımızı oluştururken ‘Cycle’ oluşuyorsa o kenarı ağaca dahil etmememiz gerekmektedir

.

## PROJE BİLGİLERİ

### Projemin Konusu

* Projemin konusu ‘Minimum Yolun Bulunması Uygulaması’ problemidir.

### Projemin Adı

.

* Kruskal Algoritması ve Mantığı’dır.

### Projemin Programlama Dili

* Projemi programladığım dil ‘C’ dilidir.

### Projemi Programladığım Platform

* Projemi programladığım platform ‘DEV C++’tır.

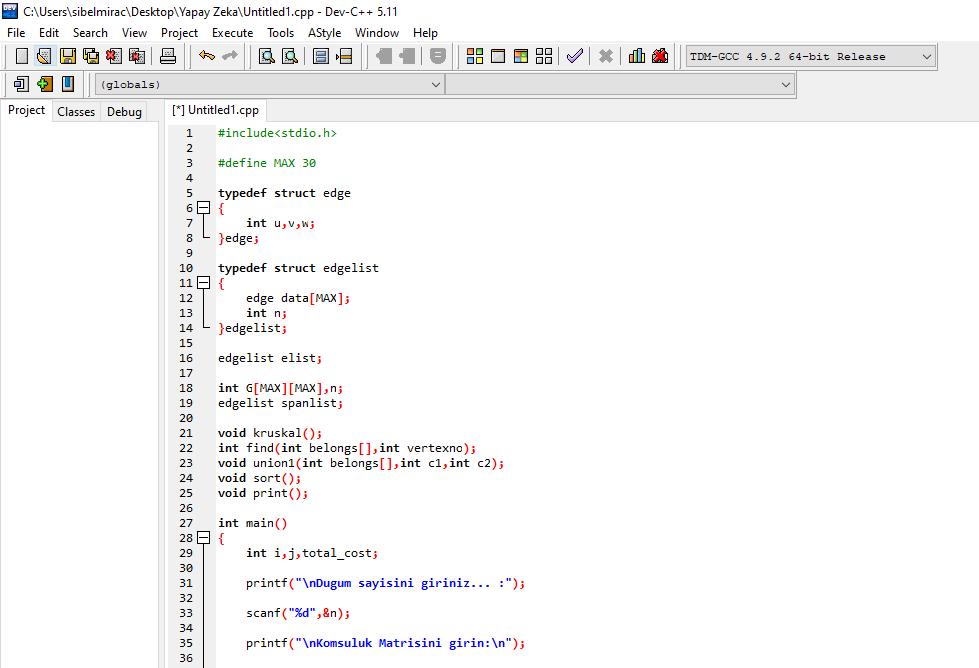
### Projemin Minimum Yolun Bulunması Uygulaması Tasarımı

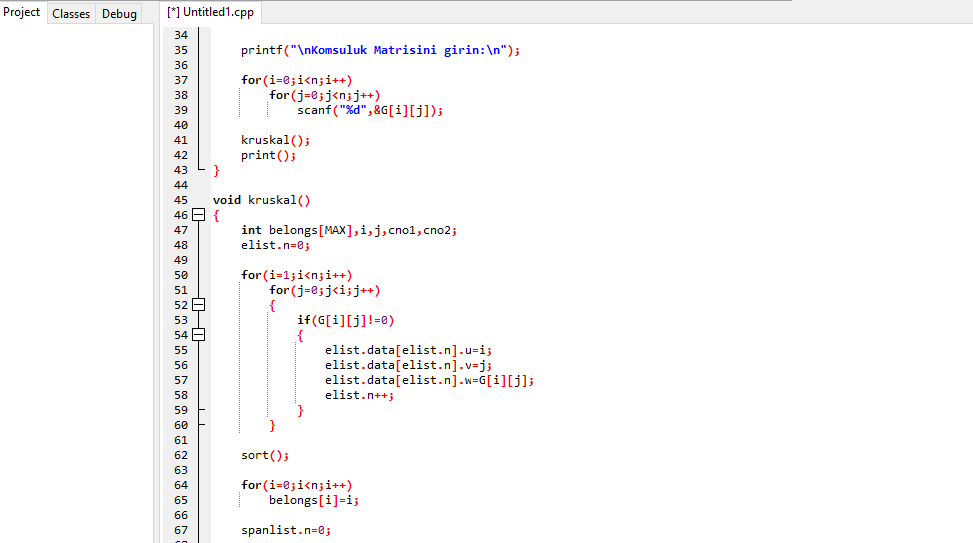
* Uygulamamın tasarımında ilk öncelikle kullanıcıdan düğüm ve matrisler istendikten sonra program kendi içinde ‘Cycle’ oluşmamasına dikkat ederek ve kenarları küçükten büyüğe sıralayarak bir asgari tarama ağacı oluşturacak ve son kısmında da en az maliyeti hesaplayarak ekrana en az maliyeti yazdıracak.

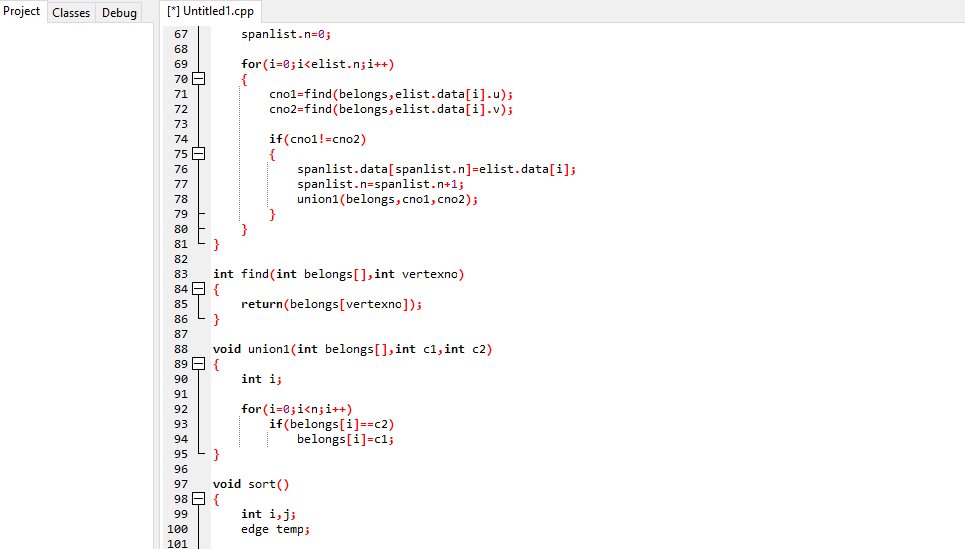
### Projemin Algoritması

* Başlangıçta kullanıcıdan düğüm girilmesi istenecek.
* İkinci aşamada girilen düğüme bağlı olarak Matris girilmesi istenecek.
* Üçüncü kısımda girilen değerlere göre kenarlar küçükten büyüğe sıralanacak.
* Dördüncü aşamada sıralanan kenarlarla bir asgari tarama ağacı oluşturulacak.
* Beşinci aşamada ‘cycle’ oluşmayan kenarların oluşturduğu tarama ağacımın maliyeti hesaplanacak.
* Altıncı ve son aşamada ise asgari tarama ağacımın en az maliyeti ekrana yazdırılacak.

### Projemin Kodlaması

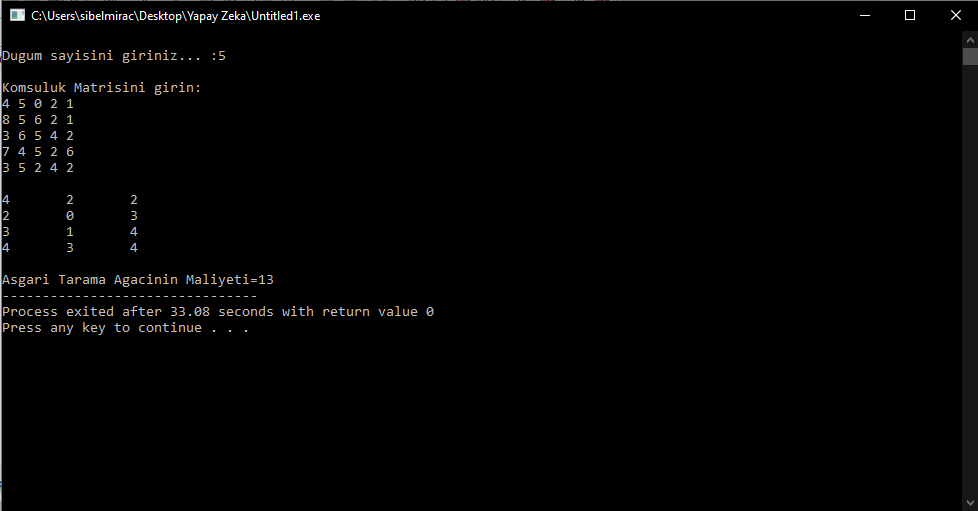








### Projemin Ürettiği Ekran Çıktıları





### Projemi Nasıl Yaptım?

* İlk öncelikle kullanıcıdan bir düğüm sayısı girilmesini istiyorum, daha sonra düğüm sayısına bağlı olarak matris girmesini istiyorum. Bu işlemleri ana fonksiyonumun içinde kodluyorum.
* Daha sonra ‘Struct’ tanımlıyorum ve structlar yardımı ile kenar yapılarımı tutuyorum. Tuttuğum yapılarım; ‘u’,’v’,’w’. Bu kısımda ki ‘u’ ve ‘v’ düğümleri temsil etmekte ve ‘w’ ise ağırlığı temsil etmektedir.
* Yine structlardan yardım alarak kenar listesi tutan bir yapı oluşturuyorum ve MAX değerini ‘30’ olarak belirtiyorum.
* ‘ void kruskal()’ olarak tanımladığım yardımcı fonksiyonumun içinde destination düğümleri ile normal düğümler arasında ki maliyetleri değişkenlere aktarıyorum. Daha sonra ‘sort()’ fonksiyonunu çağırıyorum. Bu fonksiyon kenarların küçükten büyüğe sıralanmasında rol oynamaktadır. Daha sonra yine aynı yardımcı fonksiyonum içerisinde düğümlerimin parentlarını oluşturuyorum.
* Find() fonlsiyonum ile bir öğemin hangi alt kümeye ait olduğunu belirletiyorum. Aynı zamanda birden fazla alt kümede olup olmadığı da kontrol ediliyor.
* Union() fonksiyonuyla da döngü oluşup oluşmadığını kontrol ediyorum.
* Son olarak ise print() yardımcı fonksiyonumun içinde asgari tarama ağacımın maliyetini ekrana yazdırma işlemini gerçekleştiriyorum.

## ÖZ GEÇMİŞ

* İlkokul ve Ortaokulu Pendik Merkez İlköğretim Okulunda okudum.
* Atalar Fen Bilimleri Temel Lisesinden mezun oldum.
* Üniversite hayatıma Okan Üniversitesinde devam etmekteyim.
* Mekatronik bölümünü okumaktayım ve Bilgisayar Programcılığı bölümüne Çift Anadal yapmaktayım.
* Üniversitede görmüş olduğum diller ‘C’,’C++’,’C#’,’Java’ ve ‘Kotlin’ dilleridir.
* Üniversitede görmüş olduğum programlar ise ‘Android Studio’, ‘Visual Studio’,’Code Blocks’,’Dev C++’ ve ‘Arduino’ programlarıdır.

## KAYNAKÇA

* [**http://bilgisayarkavramlari.sadievrenseker.com/2007/12/24/kruskal-asgari-tarama-agaci-algoritmasi/**](http://bilgisayarkavramlari.sadievrenseker.com/2007/12/24/kruskal-asgari-tarama-agaci-algoritmasi/)
* **https**[**://www.youtube.com/watch?reload=9&v=k1YPim47hmk**](https://www.youtube.com/watch?reload=9&v=k1YPim47hmk)
* [**https://www.stendustri.com.tr/bilim-teknoloji/yapay-zeka-nedir-ne-demektir-nasil-calisir-h99459.html**](https://www.stendustri.com.tr/bilim-teknoloji/yapay-zeka-nedir-ne-demektir-nasil-calisir-h99459.html)
* [**https://www.youtube.com/watch?v=RPHc\_IxNYdQ**](https://www.youtube.com/watch?v=RPHc_IxNYdQ)
* [**https://sites.google.com/a/mersin.edu.tr/yoneylem/ag-algoritmalari/minimum-kapsama-agaci**](https://sites.google.com/a/mersin.edu.tr/yoneylem/ag-algoritmalari/minimum-kapsama-agaci)

## EK

### Proje Kodları Ana Başlıkları

* int main()
* void kruskal()
* void union1()
* void sort()
* int find()
* void print()

## PROJEMİN YAYINLANMASI

### GitHub Linki

* Proje teslimim ‘GitHub’ sayfası üzerinden teslim edilmiştir. Link;
* <https://github.com/sibelmirac/sibelmirac>