**MULTITHREAD KULLANARAK SAMURAI SUDOKU ÇÖZME**

*Metehan MART, M. Gürkan CİNGÖZ*

*Bilgisayar Mühendisliği Bölümü Kocaeli Üniversitesi*

*metehanmart@gmail.com, gurkancingoz@yandex.com*

# **PROBLEM TANIMI**

Projede yapılması istenilenleri 3 maddede inceleyebiliriz. Samuray sudokuyu 5 thread kullanarak çözmek, samuray sudokuyu 10 thread kullanarak çözmek ve bu iki çözümün çözme hızını karşılaştıran bir grafik çizdirmek.

# **YAPILAN ARAŞTIRMALAR**

İlk önce tek bir sudokuyu çözmek için gerek algoritmalar araştırılırmıştır. Burada 1. isteri yapılmasına imkan veren backtracking algoritması kullanılmıştır.

# **YAZILIM MİMARİSİ**

C# .NET 6.0 kullanılarak kodlanmıştır. Ayrıca herhangi bir ek kütüphane kullanılmamıştır.

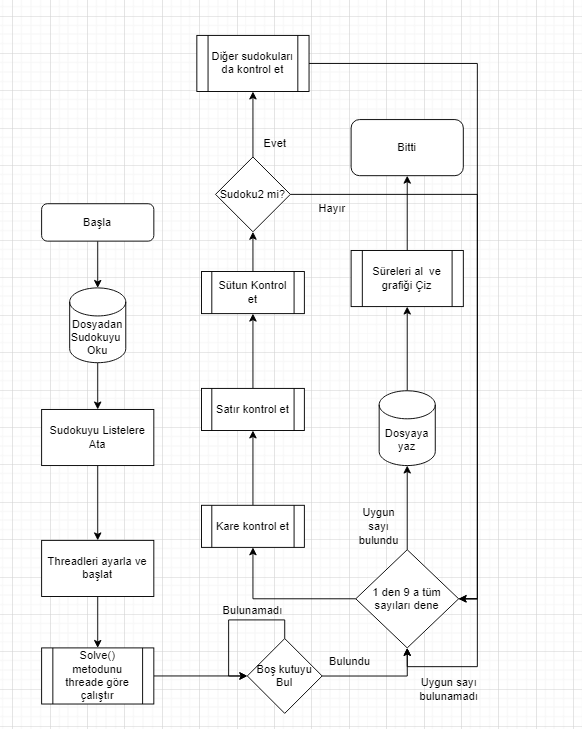
# **ER DİYAGRAMI**

Projede veri tabanı kullanılmamıştır.

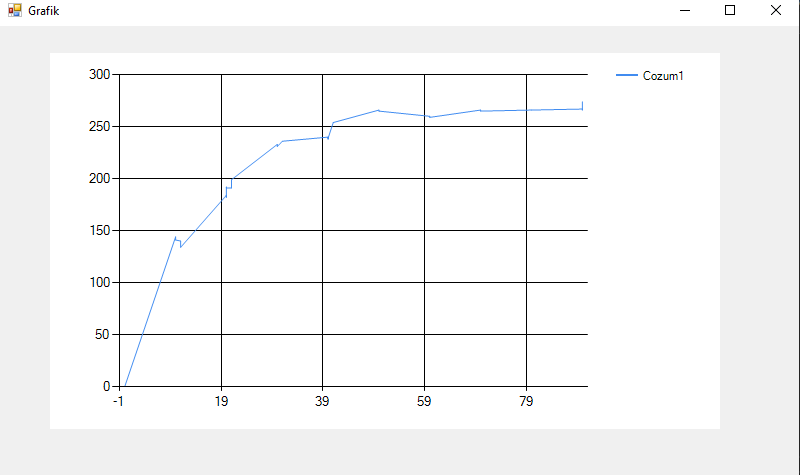
# **GENEL YAPI**

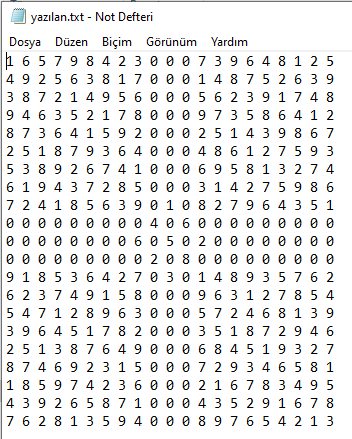
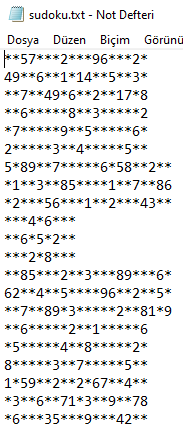
Proje program.cs dosyasındaki main methodundan çalışmaya başlıyor. Burada yeni.cs’in constructerını çağırarak başlatıyor. Yeni.cs de samuray sudoku çözülüyor. Çözümde backtraking algoritması kullanılıyor. Constructerda önce samuray sudoku verilerini txtden almamız için uygun bir array oluşturuluyor. Bunun için readTxt methodu çalışyor. Bu method arrayin içini dolduruyor. Sonra atama methodu ile aynı bir çeşit listeye dönüştürülüyor. SudokularTam değişkeni ise samuray sudokunun tamamını temsil ediyor. Constructorda her bir sudoku için thread başlatıyor. Çözme işlemi her sudoku için ayrı ayrı gerçekleşmesi böyle sağlanmıştır. Çözüm için Solver classından obje oluşturuluyor. Solver classı kendi içerisndeki solve methodunu çağıtıyor. Bu solve methodunda recursiveMethod çağrılıyor. Bu recursiveMethod hangi sudoku için çalışıyorsa o sudokuda boş yer kalmayana kadar çalışıyor. Eğer bir hücre için çözüm bulamazsa geriye dönüş başlıyor. Bir önceki hücredeki sayıyı arttırmaya çalışır ve tekrar dener böylelikle tüm hücreler dönene kadar çalışır. Solver classının içinde çözülen hücre sayısını tutulmasını sağlayan değişkenler de mevcuttur. O değişkenler uygun yerlerde arttırılıp uygun yerlerde azaltılarak çözülen hücre sayısının doğru bir şekilde tutulmasını sağlar. Bu işlemler bittikten sonra grafik.cs dosyası çalışır ve grafiği gösterir.

# **AKIŞ ŞEMASI**



# **DENEYSEL SONUÇLAR**





# **KAYNAKÇA**

Web siteleri:

<https://app.diagrams.net/> (Erişim Tarihi: 29.11.2021)

<https://dev.to/aspittel/how-i-finally-wrote-a-sudoku-solver-177g> (Erişim Tarihi: 15.11.2021) <https://www.geeksforgeeks.org/sudoku-backtracking-7/> (Erişim Tarihi: 15.11.2021) <http://www.albahari.com/threading/> (Erişim Tarihi: 15.11.2021)

<https://www.youtube.com/watch?v=gqo2TGpCOlA&ab_channel=FoxLearn>(Erişim Tarihi: 20.11.2021)

<https://stackoverflow.com/questions/4016483/get-time-in-milliseconds-using-c-sharp/4016511> (Erişim Tarihi 25.11.2021)