

## Yazılım Laboratuvarı I

# PROJE#2 MULTİ THREAD KULLANARAK SAMURAI SUDOKU ÇÖZME

Proje Teslim Tarihi : 29.11.2021

### Problemin Tanımı

Bu projede .txt uzantılı dosya içerisinde başlangıç değerleri verilecektir ve bu değerlere göre samurai sudokuyu oluşturmamız, sonrasında da bu samurai sudokuyu çözmemiz istenmektedir. Çözüm işlemlerini ise multi thread yapısını kullanarak gerçekleştirmemiz gerekmektedir. Çözüm işlemleri gerçekleşirken tüm threadlerin işlem adımlarının .txt uzantılı dosya veya veritabanında tutulması istenmektedir. Samurai sudonunun çözülme aşamalarını sırayla form ekranında göstermemiz gerekmektedir. Samurai sudokunun çözümü beş ve on thread olarak iki farklı şekilde yapılacaktır, bu çözümlerin bulunduğu kare sayılarını ve çalışma sürelerini grafik üzerinde göstermemiz istenmektedir.

### Yapılan Araştırmalar

Öncelikle .txt uzantılı dosyadan okuduğumuz verileri nasıl bir yapıda tutmamız gerektiğini araştırdık bunun sonucunda bir adet yirmi bir elemanlı düzensiz dizi oluşturduk. İlgili verileri form ekranında gösterebilmek için aynı tipte bir

buton nesnesi daha oluşturduk. Bundan sonraki işlemde bir sudokunun hangi adımlarla çözülebileceğine dair araştırmalar yaptık, bunun için backtracking kavramını kullandık. Beş ayrı sudokunun aynı anda çözüme başlaması için derinlemesine bir thread araştırması yaptık. Loglama işlemi için .txt uzantılı dosya oluşturmaya karar verdik ve streamwriter fonksiyonunu kullandık. Grafik çizdirme işlemi için windows formda bulunan chart özelliğini kullandık, buraya işlenecek verilerin süre hesabı için stopwatch kullanılmıştır.

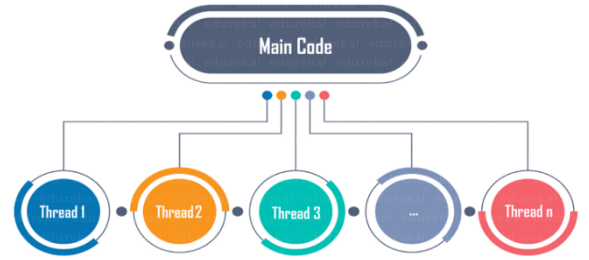
### Genel Yapı ve Yöntem

Projeye ilk olarak sudoku.txt dosyasında bulunan değerleri okuyarak başladık. Bunun için filestream fonksiyonu ile sudoku.txt uzantılı dosyanın yolunu ve yapacağımız işlemin belirlenmesi sağlanmıştır. Dosyanın sonuna gelene kadar mevcut satırdaki tüm veriler bir dizi aracılığıyla tutulmuştur. Sonrasında bu verilerin ekranda gösterilebilmesi için form uygulaması üzerinde rakamların denk geleceği butonlar oluşturulmuştur. Bu butonların textlerine dizide tuttuğumuz değerler aktarılmıştır. Kullanıcının sudoku çözümü için tek thread, beş thread ve on thread işlemlerinden seçim yapabilmesi, çoklu threadlerin çalışma zaman-çözüm karesi grafiğini göstermek için ilgili butonlara da yer verilmiştir. Her bir sudokunun çözümü için sırayla birden ona kadar rakamların ilgili butona uyup uymadığını kontrol eden algoritma tasarlanmıştır, bu algoritma rakam koyulacak butonun ilgili satırında, sütununda ve 3x3'lük karesinde kontrol edilen rakamın bulunup bulunmadığı sorgulanmıştır. Eğer mevcut rakam o butona yazılabiliyorsa o butona bu rakam aktarılır ve bir sonraki buton için aynı adımlar tekrarlanır ama ilgili buton için hiçbir

rakam uymuyor ise kullandığımız backtracking yöntemiyle daha önceki butonlara yerleştirdiğimiz rakamlarda değişikliğe gidilmesi sağlanmıştır. Tüm butonlar doldurulmuş ise çözüme ulaşılmış demektir. Samurai sudokunun çözümü için ortada bulunan sudoku bir çözüme ulaştırılır ve ortadaki sudokunun diğer sudokularla kesişen kareleri ilgili sudokulara aktarılır. Diğer sudokuların bu aktarılma işleminden sonra çözüme ulaşması beklenilir eğer bir çözüm ortaya çıkmaz ise ortadaki sudoku için başka bir çözüm yolu aranır ve bu durum tüm sudokular çözülene kadar devam eder. Beş threadli çözüm için ortadaki sudokunun çözümünün ardından thread işlemlerinde bulunan start metodu ile tüm sudokular aynı anda çözüme başlar ve tüm butonların doldurulup doldurulmadığını kontrol etmek için join metodu ile tüm thread işlemlerinin bitmesi beklenir. On threadli çözüm içinse aynı işlemler uygulanır fakat her bir sudokun bir adet başında ve bir adet sonunda olmak üzere iki adet thread bulunur. Tüm işlem adımlarını bir .txt uzantılı dosya aracılığıyla sudoku üzerindeki eklenen ve silinen her bir rakamın loglanma işlemleri yapılmıştır. Program her çalıştırıldığında .txt uzantılı dosya oluşturulur ve ismi çalıştırılma anındaki sistem saati olur. Grafiğin oluşturulması için ilk etapta samurai sudoku beş threadle çözüme ulaştırılır sonrasında sudoku resetlenip on threadle tekrardan çözüme ulaştırılır. Bu iki durumdaki çözümlerde bulunan kare sayısı ve karelerin bulunma süresi bir arraylist aracılığıyla tutulup bu veriler grafikte gösterilir. Grafik çizdirme işlemi içinse windows formda bulunan chart özelliğini kullanılmıştır.

## Yazılım Mimarisi

Tasarlanan sistem multi thread yapısından oluşmaktadır. Bu multi thread yapısını single thread yapısından ayıran faktörlerden en önemlileri; single thread bir seferde tek bir iş parçacığı çalıştırırken multi thread birden fazla iş parçacığını aynı anda çalıştırır, single thread'in boşa kalma süresi daha fazlayken multi thread'de bu süre daha azdır, multi thread kullanmak daha verimli programlarla sonuçlanır, single thread yapısında bir iş parçacığı duraklatıldığında sistem bu iş parçacığı devam ettirilene kadar bekler ama multi thread yapısında diğer iş parçacıkları normal şekilde çalışır.



## Referanslar

### Web Site

- <https://docs.microsoft.com/tr-tr/dotnet/api/system.windows.forms.datavisualization.charting.chart?view=netframework-4.8>
- <https://www.gencayildiz.com/blog/c-ile-multi-thread-programlama-temelleri/>
- [https://en.wikipedia.org/wiki/Sudoku\\_solving\\_algorithms](https://en.wikipedia.org/wiki/Sudoku_solving_algorithms)
- <https://www.geeksforgeeks.org/backtracking-introduction/>

## Akış Şeması

