

SAMURAI SUDOKU PROJESİ

RECEP KOCA

MUHAMMED FURKAN BEYAZ

Mühendislik Fakültesi Bilgisayar Mühendisliği

KOCAELİ ÜNİVERSİTESİ

kocarecep95@gmail.com

furkanve747@gmail.com

1.PROJE TANITIM

Sudoku mantık bazlı , birleştirici sayı yerleştirme bulmacasıdır. Klasik sudokuda ama. , 9*9 matrisi rakamlarla doldurmaktır. Böylece her sütun , her satır ve matrisi oluşturan dokuz 3*3 alt matrisin her biri 1'den 9'a kadar olan tüm rakamları içerir. Bulmaca ayarlayıcı , iyi hazırlanmış bir bulmaca için tek bir çözümü olan , kısmen tamamlanmış bir matris sağlar. Sumari sudoku bulmacaları , örtüşen beş sudoku matrisinden oluşur. Standart sudoku kuralları , her 9*9 matris için geçerlidir. Samurai sudokudaki bulmacalar , tahmin etmeye gerek kalmadan saf mantıkla bulunabilecek benzersiz bir çözüme sahiptir. Bu projede ise multithread yapısını kullanarak verilecek samurai sudoku üzerinden çözüm bulmanız istenmektedir. Sudoku başlangıç değerleri .txt uzantılı dosyada verilecektir. Verilen değerler dinamik olarak uygulamaya aktarılacaktır. Birinci ister olarak verilen samurai sudoku içindeki her bir sudoku için bir başlangıç noktası seçerek 5 thread ile çözüme ulaşmamız beklenmektedir. Threadler eşzamanlı çözüme başlayacaktır. Bu sudoku içerisinde senkronizasyon problemi göz önünde bulundurulmalıdır.İkinci ister olarak verilen samurai sudoku içindeki her bir sudoku için 2 tane başlangıç noktası seçerek 10 thread ile çözüme ulaşmamız beklenmektedir. Üçüncü ister olarak ise bu 2 ister arasında zaman ve bulduğu çözüm karesi arasında ilişki grafiği çizdirilecektir.

2.GİRİŞ

Projede algoritma yeteneğinin geliştirilmesi, thread yapısının anlaşılması amaçlanmıştır.Txt dosyalarından sudoku alınıp tekrar txt ye

hamlelerin yazılması ve adımların gösterilmesi gerekmektedir. Proje JAVA dili kullanılarak yazılmıştır. Geliştirme ortamı olarak Netbeans IDE kullanılmıştır.

3.TASARIM ve YAZILIM MİMARİSİ

Uygulamada 5 class yapısı kullanılmıştır. 3 tane Java class , 2 tane de JFrame classından oluşmaktadır. Hamle classında satır sütun değişkenleri tanımlanıp getter setter metodları oluşturulmuştur. SudokuThread classında hamleleri tutan bir arraylist ve süreyi tutmak için değişkenler tanımlanmıştır. Ardından run metoduyla threadler farklı yerlerden çözüm işlemlerine başlamaktadır. Sudokuyu çözmeye kodları yazılıp çözülen sodoku ekrana bastırılmıştır ve son metod ise hamleleri dosyaya yazmaktadır. TextGrafik classı JFrame classıdır. Basit bir design oluşturulmuştur. Design kısmında txt leri alıcak butonlar oluşturulmuştur. Butonların altında ise txt den okuduğumuz değerleri yazacak textarea bulunmaktadır.Graphic classı da JFrame classıdır. Bu classta çözülen kareye göre süre grafiği çizdirilmesi beklenmektedir. Son olarak ise main classı bulunmaktadır. Bu classta dosyaya yazma ve dosyadan okuma işlemi yapılmaktadır. 5 adet sudoku oluşturulmuştur ve SudokuThread classından nesne oluşturulup bu sudokular sudokuthreada parametre olarak aktarılmaktadır. Son olarak jframe classları çağırılıp sudokular görsel olarak ekrana bastırılmaktadır.

4.YAPILAN ARAŞTIRMALAR

Multithread yapısını udemy ve youtube dan dersler izleyerek anlamaya çalıştık. Sudoku için kaynağı ise yabancı kaynakları araştırarak elde etmeye çalıştık. Araştırmalarımızı daha çok internet üzerinden yabancı kaynaklar bularak projede gerçeklemeye çalıştık.

4.PROBLEM TANIMI VE YÖNTEM

Projeye başlamadan önce sudoku oyununu tam anlamıyla çözmek ve algoritma geliştirmek için oynamaya başladık. Samurai Sudoku oyununu ise ilk defa duymuştuk. Ardından sudokunun bu çeşidini nasıl çözeriz diye bunu da oynamaya başladık. Ancak bu oyunu oynarken bile zorlanıyorduk. Samurai Sudoku için 5 tane sudoku oluşturduk. Ancak bunları birleştiremedik. Samurai sudokunun algoritmasını oluşturamadık. Bu yüzden 5 tane ayrı sudokuyu threadler yardımıyla çözdürmeye karar verdik. Multithread ilk önce bize uzak bir kavram gibi görünüyordu. Udemyden multithread dersleri almaya başladık. Birkaç gün bu dersleri dinledik ve anlamaya çalıştık. Ardından projeyi geliştirmek için hangi dille yazacağımızın kararını verdik. JAVA diliyle yazmaya karar verdik. Oluşturduğumuz sudokuları txt ye okuma işlemi yaptık ve hamle adımlarını tek tek gösterdik. Ardından GUI kısmına geçtik. GUI de bir adet menü oluşturduk. Menüde butonlar koyup oradan hangi butona tıklanıyorsa o txt den veriyi çekme işlemi gerçekleştirdik. Veri çekme işleminde ise JTextArea oluşturduk. JTextArea kısmında txt den alınan verileri GUI ekranına bastırdık. Diğer GUI de ise grafik ekranı bulunmaktadır. Main classında ise SudokuThread de oluşturulan threadler başlatılıyor. Sudoku tahtası bu threadlere aktırılıyor. Ardında 1.GUI classından nesne oluşturulmuştur. Ve setvisible fonksiyonuyla etkin hale getirilmiştir.2 GUI classının nesneside main classında oluşturulup main classında çağırılmıştır.

5.DENEYSEL SONUÇLAR

```
run:
Çözülecek Sudoku
-----
| * * 5 | 7 * * | * 2 * |
| 4 9 * | * 6 * | * 1 * |
| * * 7 | * * 4 | 9 * 6 |
-----
| * * 6 | * * * | * * 8 |
| * 7 * | * * * | * 9 * |
| 2 * * | * * * | 3 * * |
-----
| 5 * 8 | 9 * * | 7 * * |
| * 1 * | * 3 * | * 8 5 |
| * 2 * | * * 5 | 6 * * |
-----
```

çözülecek ikinci sudoku

```
-----
-----
| * * 9 | 6 * * | * 2 * |
| 1 4 * | * 5 * | * 3 * |
| * * 2 | * * 1 | 7 * 8 |
-----
| * * 3 | * * * | * * 2 |
| * 5 * | * * * | * 6 * |
| 4 * * | * * * | 5 * * |
-----
| 6 * 5 | 8 * * | 2 * * |
| * 1 * | * 7 * | * 8 6 |
| * 2 * | * * 4 | 3 * * |
-----
```

sudoku Cozucu 2 tarafından 94 milisaniyede çözülmüştür

```
-----
| 7 3 9 | 6 4 8 | 1 2 5 |
| 1 4 8 | 7 5 2 | 6 3 9 |
| 5 6 2 | 3 9 1 | 7 4 8 |
-----
| 9 7 3 | 5 8 6 | 4 1 2 |
| 2 5 1 | 4 3 9 | 8 6 7 |
| 4 8 6 | 1 2 7 | 5 9 3 |
-----
| 6 9 5 | 8 1 3 | 2 7 4 |
| 3 1 4 | 2 7 5 | 9 8 6 |
| 8 2 7 | 9 6 4 | 3 5 1 |
-----
```

```

sudoku Cozucu 3 tarafından 110 milisaniyede çözülmüştür

-----
| 1 6 5 | 7 9 8 | 4 2 3 |
| 4 9 2 | 5 6 3 | 8 1 7 |
| 3 8 7 | 2 1 4 | 9 5 6 |
-----
| 9 4 6 | 3 5 2 | 1 7 8 |
| 8 7 3 | 6 4 1 | 5 9 2 |
| 2 5 1 | 8 7 9 | 3 6 4 |
-----
| 5 3 8 | 9 2 6 | 7 4 1 |
| 6 1 9 | 4 3 7 | 2 8 5 |
| 7 2 4 | 1 8 5 | 6 3 9 |
-----

```

6.SONUÇ

Bu projede multithread mantığını anlamaya çalıştık.Bunu eğitimler olarak yaptık. Ancak tam olarak threadleri uyguladık diyemeyiz. Sudoku koduna uyarlamakta sıkıntılar yaşadık. Samurai sudokunun ise algoritmasını kuramadık. Yeni bir çözüm yolu aramaya başladık ve klasik sudoku üzerinde thread işlemleri yaptık. Bu proje algoritma yeteneğini bir sonraki seviyeye çıkarmakla beraber thread işlemlerini ve buna bağlı olarak multithread kavramını öğrenmemize yardımcı oldu.

7.REFERANSLAR

<https://www.udemy.com/home/my-courses/learning/>

<https://practice.geeksforgeeks.org/problems/solve-the-sudoku-1587115621/1>

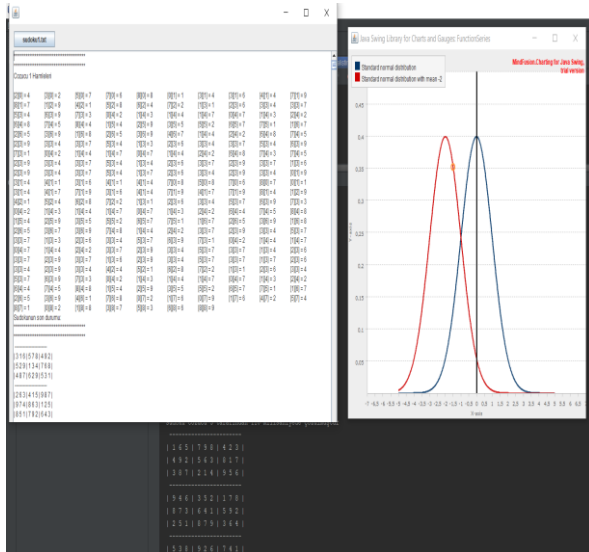
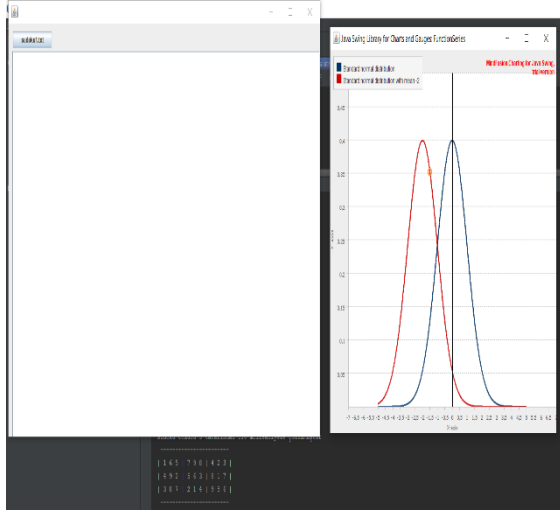
<https://www.baeldung.com/java-sudoku>

<https://www.samurai-sudoku.com/>

<https://www.codeproject.com/Articles/237372/Informed-search-algorithms-to-solve-Sudoku-Samurai>

https://www.tutorialspoint.com/java/java_multithreading.htm

<https://www.geeksforgeeks.org/multithreading-in-java/>



8.ALGORİTMA ŞEMASI

