

MULTITHREAD KULLANARAK SAMURAI SUDOKU ÇÖZME

Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

Cihan İçelliler

icellilercihan@gmail.com

Gülşay Genç

gulaygenc08@gmail.com

I. PROBLEM TANIMI

Bu projede amaç thread kullanarak aynı anda senkronize bir şekilde samurai sudokusunu sırasıyla 5 thread ve 10 thread ile çözdürmektir.

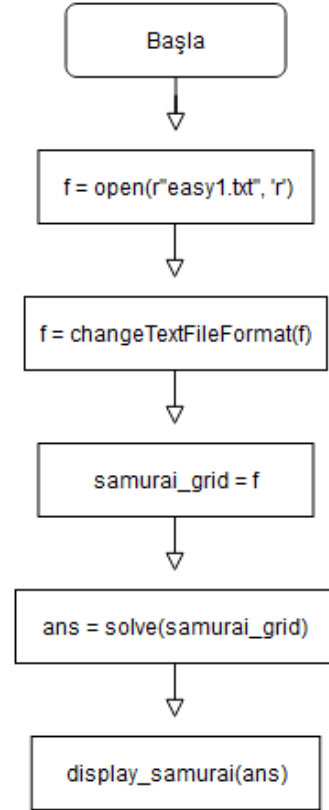
II. YAPILAN ARAŞTIRMALAR

Projemizde istenen samurai sudoku çözümünü yapmak için kaynak bakımında zengin olması sebebiyle araştırıp seçtiğimiz dil Python oldu. Bunun için proje kodları yazılırken Visual Studio kullanıldı.

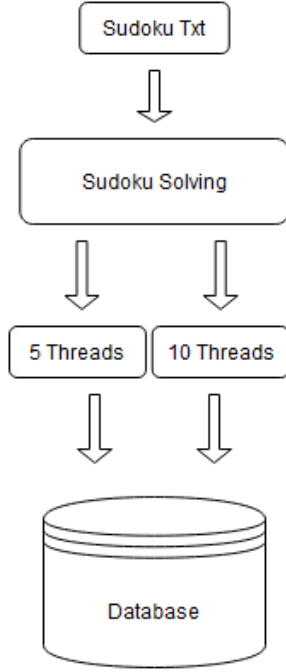
Sudoku'nun çözümü sırasında gösterilecek grafik için ise gui tasarımında Pygame kütüphanesi kullanıldı. Kodumuzu yazarken birçok platformdan örnek kodlar araştırdık ve bulduklarımızı projemize uyarladık.

Son olarak çözüm yapıldıktan sonra gösterilmesi istenen grafik için matplotlib kütüphanesini kullanmaya karar verdik.

III. AKIŞ ŞEMASI

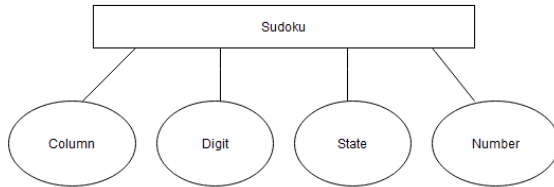


IV. YAZILIM MİMARİSİ



Tasarlanan sistemde sudoku için alınacak başlangıç değerler txt'den alınmaktadır. Txt içerisinde değerler alınıp sudokuya yerleştirilmektedir. Sudoku çözümü 5 ve 10 thread'li olarak yapıldıktan sonra her sudokunun kendine ait ID'si ile satır ve sütun bilgisi yardımıyla son değerler tutulmaktadır ve değerler database'e yazılmaktadır.

V. VERİ TABANI DİYAGRAMI



Database olarak csv dosyası kullanılmıştır. Değerler column(çözülen sütun), digit(çözülen satır), state(hangi sudoku olduğu) ve number(çözüm) olarak değişkenler yardımıyla tutulmuştur.

```
Column and Digit,State,Numbers
G4,b,8
B4,a,5
C3,a,7
D5,d,7
H7,a,2
F9,b,3
E6,d,9
A4,a,7
```

VI. GENEL YAPI

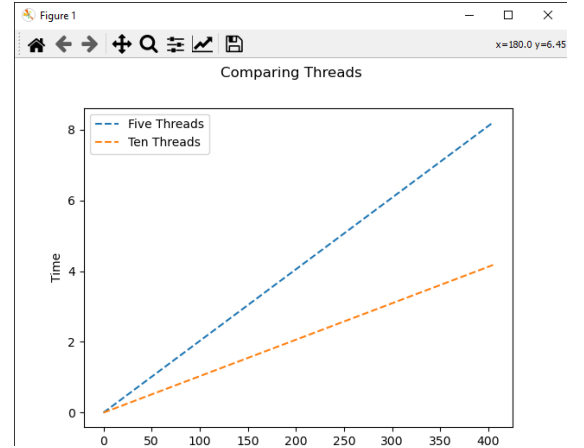
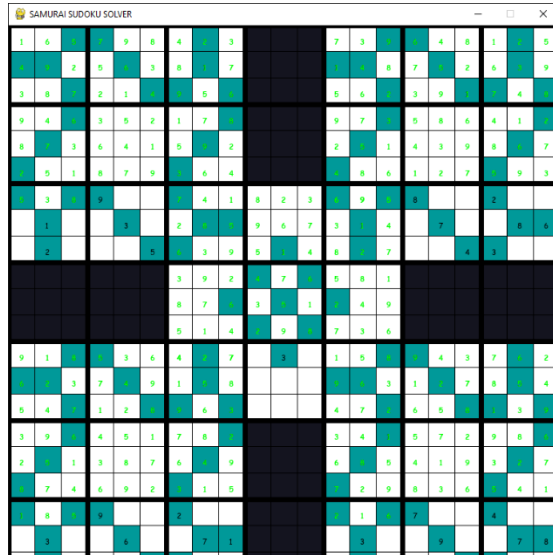
Köşelerde yer alan sudokulara a'dan d'ye ID'ler verilmiştir, ortadaki 5. sudokumuza "+" ID'si verilmiştir. Satırlar rakamlarla, sütunlar ise harflerle kodlanarak cross() fonksiyonuyla kartezyen çarpımı yapılarak her hücreye bir ID üretilmiştir. Ortadaki sudokumuzun diğer köşelerde yer alan sudokularla kesişen kısımların kontrolü için repl() fonksiyonu kullanılmıştır bu sayede daha önce ID verilmiş hücreye tekrar farklı bir ID atanması önlenmiştir.

Open() fonksiyonuyla okunan txt içerisinde değerleri almak için changeTextFileFormat() fonksiyonu kullanılmıştır. Burdan alınan değerler solve() fonksiyonuna gönderilmiştir. Bu fonksiyon alınan değerleri parse_grid_samurai() fonksiyonuna göndererek atanan satır ve sütunlara yerleştirilmesini sağlamıştır. Burdan sonra search() fonksiyonuyla atanan değerlerin kontrolü ve boşluklara gelebilecek eksik değerlerin hesaplanması yapılmıştır, değerlerde herhangi bir hata alınmadığı takdirde write_points_to_database() fonksiyonuyla boşlukların alabileceği değerler Points adlı csv dosyamıza yazılır, son çözüm değerleri ise PointsSuccesss adlı csv dosyamıza yazılmıştır. Sudokuların çözülmüş hali display_samurai() fonksiyonuyla tek tek çıktı olarak verilmiştir.

Gui için pygame ile bir ekran oluşturulmuştur. Ekran üzerinde sudokunun çizimi draw_box() ve draw() fonksiyonları kullanılarak yapılmıştır. Her bir sudokunun hem tek thread hem iki threadli olarak çözümü sırasında time bileşeni kullanılarak aktif olarak denenen çözümlerin de görülmesi sağlandı.

Son olarak istenen geçen zaman ve bulunan çözüm karesi sayısı arasındaki ilişkinin gösterimi için matplotlib kütüphanesi kullanılarak bir grafik oluşturulmuştur. Geçen zamanın bulunması için her iki çözüm için de ayrı ayrı olarak time() fonksiyonu kullanılarak başlangıç ve bitiş arasındaki zaman alınmıştır. Çözüm karesi sayısı ise çözüm sırasında
founded_squares_for_five_thread ve
founded_squares_for_Ten_thread
değişkenleriyle tutulmuştur.

VII. EKRAN GÖRÜNTÜLERİ



VIII. KAYNAKÇA

<https://www.techwithtim.net/tutorials/python-programming/sudoku-solver-backtracking/>

<https://www.pygame.org/news>

<https://www.pygame.org/docs/>

<https://www.pygame.org/wiki/tutorials>

<https://matplotlib.org/>

<https://matplotlib.org/stable/tutorials/index.html>

https://matplotlib.org/stable/api/as_gen/matplotlib.pyplot.html

<https://realpython.com/intro-to-python-threading/>

<https://docs.python.org/3/library/threading.html>

https://www.tutorialspoint.com/python/python_multithreading.htm