T.C.
FIRAT ÜNİVERSİTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

YAZILIM MÜHENDİSİLİĞİ GÜNCEL KONULAR DERSİ GÜVENLİ DİJİTAL HAYAT PROJESİ DOKÜMANTASYONU

Tunahan GÖKÇİMEN

HAZİRAN-2022

İçindekiler

Şekiller listesi

Tablolar listesi

1. GİRİŞ

Proje kapsamında Mersenne Twister algoritması ile 1 milyon bit üretilmiştir. Üretilen bitlerin NIST(istatistiksel test süiti) testinden geçirilmiştir.

1.1.Mersenne Twister

Makoto Matsumoto ve Takuji Nishimura tarafından 1997 yılında geliştirilen Mersenne twister algoritması yukarıda bahsedilen pek çok problemden uzak güçlü bir sözde rassal sayı üretecidir. Bu üretecin periyodu 219937-1'dir ve 32 bitlik değerler için 623 boyutta eşdağılımlı olduğu ispatlanmış olup pek çok üreteçten daha hızlı çalışmaktadır. Bu algoritma bir süredir istatistik simülasyonlarında ve üretimsel modellemede tercih edilen rassal sayı üreteci olmaya başlamıştır.

Bununla birlikte Mersenne Twister algoritmasının çıktısını analiz edip sayıların rassal olmadıklarını anlamak mümkündür (Berlekamp-Massey algoritmasında veya bunun genişletilmiş hali olan Reed-Sloane algoritmasında olduğu gibi). Mersenne Twister algoritmasının bu bilinen olumsuz yönleri şimdilik onu şifrebilim uygulamalarında kullanılamaz kılmakla birlikte başka açılardan sorun teşkil etmemektedir (modelleme, simülasyon, vb. alanlarda kullanılabilmektedir).

1.2.NIST(istatistiksel test süiti)

- NIST istatistiksel test süiti hipotez test tabanlı bir test türüdür.
- Bu hipotez testler üretilen 0 veya 1 sayısının rastgele olup olmadığını belirler.
- Bu amaç için NIST test süitinde iki önemli parametre vardır bunlar sırasıyla α ve P-Değeri değeridir.
- Önem seviyesi olarak bilinen α ‘nın 0.01 olarak seçilmesi test edilecek sayıların rastgeleliğinin 99% güven değerine sahip olduğunu belirtir.
- Diğer parametre P-değeri rastgeleliğin ölçüsü olarak bilinir. Eğer P-değeri 1’e eşit olursa sayılar mükemmel rastgeleliğe sahiptir denir.
- P-değeri sıfıra eşit olursa sayıların rastgeliğinden söz edilemez.
- Kriptografik uygulamalarda kullanılmak üzere üretilen sayıların önem seviyesi α, uygun bir değer seçilmelidir.
- Her bir test için eğer P-değeri, α değerinden büyük ve eşit olursa test başarılıdır. Aksi durumda test başarısız yani üretilen sayılar rastgele değildir. Genellikle önem seviyesi [0.001, 0.01] aralığında seçilir.
- NIST testi güçlü bir testtir ve bu nedenle kriptografik uygulamalarda tercih edilmektedir.
- Rastgele sayı üreteçleri tarafından üretilen uzun ikili bit dizilerinin rastgeleliğini ölçmek için geliştirilmiştir.
- NIST 800-22 kendi içinde 15 tane ayrı testten oluşur.
- Teste tabi tutulan bit dizisinin başarılı olabilmesi için tüm testleri başarıyla geçmesi gerekmektedir.
- Bu testler dizi içerisindeki rastgele olmayan durumlara odaklanır.

2. MATERYAL VE METOT

Bu proje Visual Studio IDE üzerinde C# programlama dilinde geliştirilmiştir.

2.1.Rastgele Sayı Üretecinin Geliştirilmesi

Mersenne Twister algoritması kullanılarak üretilen rastgele sayı üreteci ile 1.000.000 – bit elde edilmiştir elde edilen bitler System.IO kütüphanesinin StreamWriter sınıfı kullanılarak .txt uzantılı dosyaya yazdırılımıştır.

2.1.1. Visual Studio IDE

Microsoft Visual Studio, Microsoft tarafından geliştirilen bir tümleşik geliştirme ortamıdır.

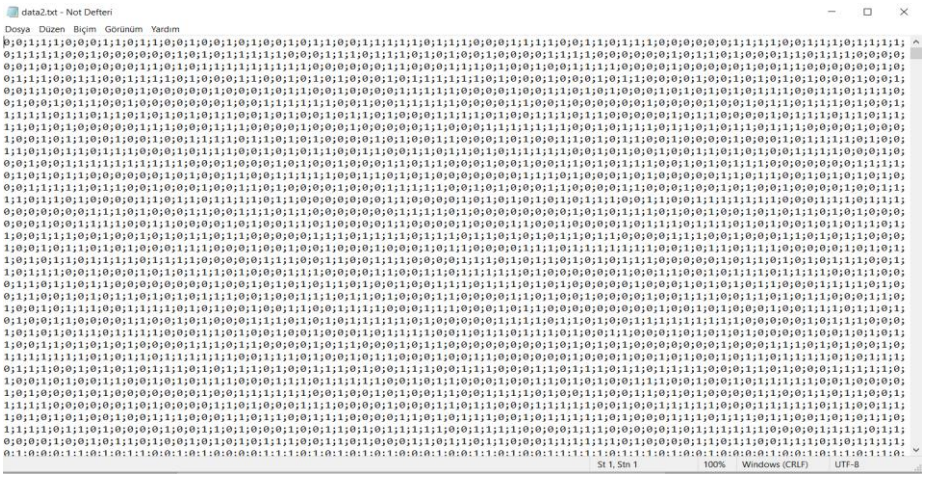
2.1.2. StreamWriter sınıfı

StreamWriter sınıfında System.IO kütüphanesi kullanılır. Bu sınıfın temel amacı metin dosyalarına karakter girişi sağlamaktır. Bilgisayar üzerinde bulunan bir dosyaya StreamWriter sınıfıyla erişilir ve bu dosyaya karakter girişi sağlanır.

2.2.Rastgele sayı üreteci ile 1.000.000 – bit üretme işlemi

```
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace RandomNumberGenerator
{
    class Program
    {
        static StreamWriter Yaz = null;
        static void Main(string[] args)
        {

            try
            {
                Yaz = new StreamWriter(@"C:\Users\TUNAHAN\source\repos\RandomNumberGenerator\data2.txt");
                for (int i = 0; i < 1000000; i++)
                {
                    Yaz.Write(RandomNumberGenerator.Instance.Generate(0, 1) + ";");
                }

            }
            catch (Exception ex)
            {
                Console.WriteLine(ex);
                Console.ReadLine();
            }
            finally
            {
                Yaz.Flush();
                Yaz.Close();

            }
```

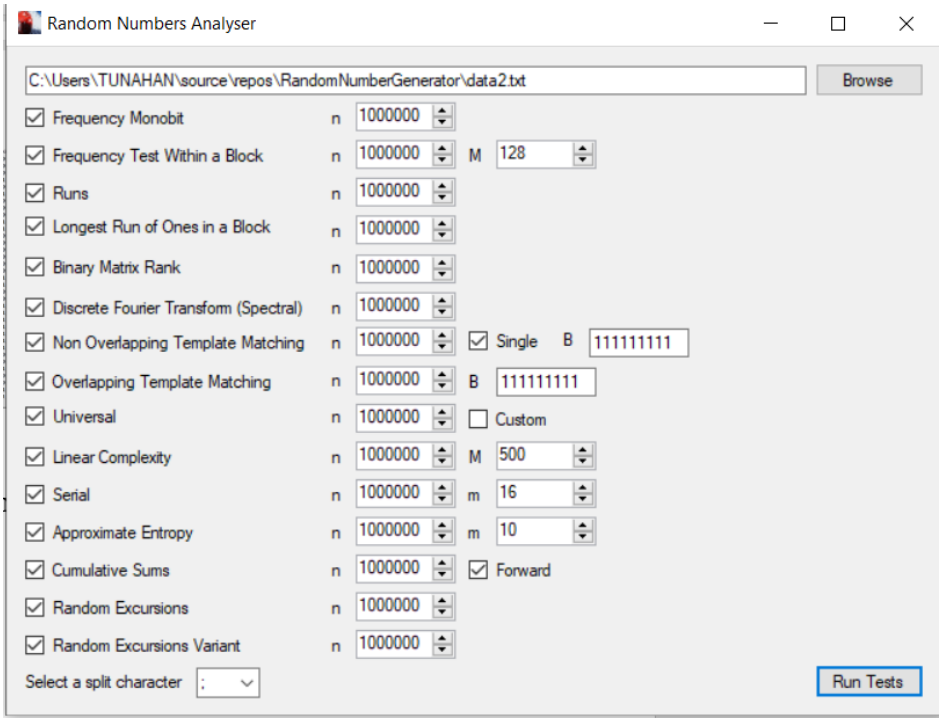
Şekil 2.1 Rastgele sayı üreteci ile 1.000.000 – bit üretme işlemi



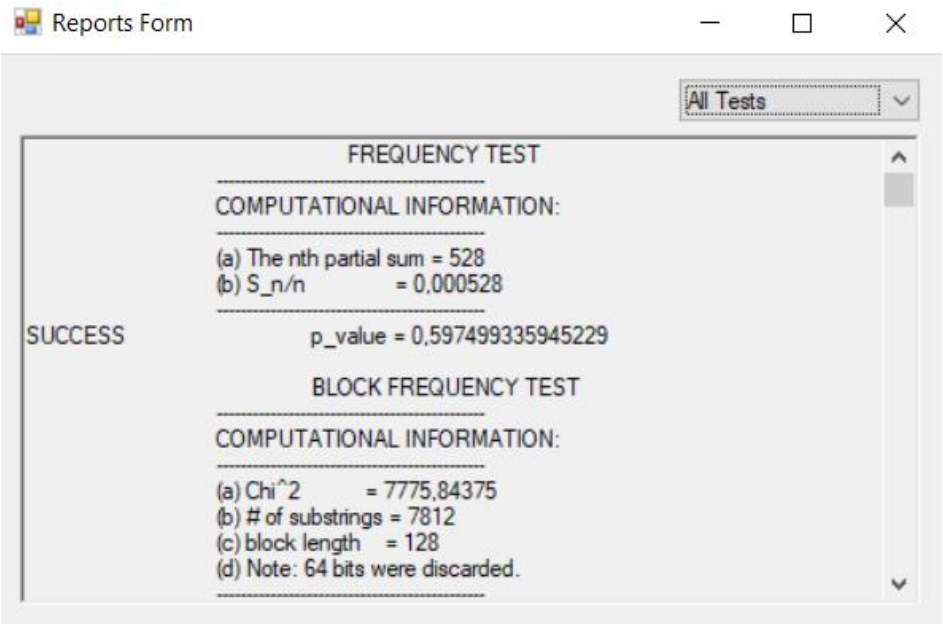
Şekil 2.2 Üretilen 1.000.000 – bit .txt dosyası görseli

2.3. ÜRETİLEN 1.000.000 – BİTE NIST TESTİ UYGULANMASI

.txt uzantıı dosyanın yolu Random numbers Analyser programında seçilerek uygulanacak Nist testleri belirlendikten sonra run test denilerek test işlemi gerçekleştirilmiştir.



Şekil 2.3 NIST testi Uygulanması



Şekil 2.4 NIST test sonuçları

3. BULGULAR TARTIŞMA

Tablo 3.1 NIST test sonuçları

	Veri
frekans testi	1
blokataki en uzun birler testi	1
örtüşmeyen şablon eşleştirme testi	1
doğrusal karmaşıklık testi	1
birikimli toplamlar testi	1
blok frekans testi	1
rank testi	1
örtüşen şablon eşleştirme testi	1
seri testi	1
rastgele gezinim testi	1
akış testi	1
ayrık fourier dönüşüm testi	1
maurer's evrensel testi	1
yaklaşık entropi testi	1
rastgele gezinim değişken testi	1
Toplam Başarı Sayısı	15

1’ er milyonluk veri setinin NIST test sonuçları Tablo 2.1 de gösterilmektedir.

Tablo 3.2 NIST test sonucu P değerleri

Testler	Veri
frekans testi	p_value = 0,5974
blokataki en uzun birler testi	p_value = 0,6119
örtüşmeyen şablon eşleştirme testi	p_value = 0,7792
doğrusal karmaşıklık testi	p_value = 0,1055
birikimli toplamlar testi	p_value = 0,3099
blok frekans testi	p_value = 0,1658
rank testi	p_value = 0,48013
örtüşen şablon eşleştirme testi	p_value1 = 0,6371
seri testi	p_value = 0,0236
rastgele gezinim testi	p_value = 0,5657
akış testi	p-value = 0,2284
ayrık fourier dönüşüm testi	p-value = 0,3239
maurer's evrensel testi	p-value = 0,2830
yaklaşık entropi testi	p-value = 0,2339
rastgele gezinim değişken testi	p-value = 0,4200

Elde edilen P değerleri Tablo 2.2 de gösterilmektedir önem değeri olan 0,01‘ den büyük olduğu için testleri başarıyla geçmiştir.

4. SONUÇ

Tüm testleri başarılı olarak tamamlayan veri setlerinin rastgeleliği kanıtlanmış olup verilerin güvenliği için şifreleme algoritmalarında anahtar olarak kullanılabileceği sonucuna ulaşılmıştır.