HOMEWORK 3 REPORT

```
while(line != null && !line.equals("")){

    String theDigits[] = line.split(" ");
    for(int i = 0; i < theDigits.length; i++){
        digits[actualLine][i] = Integer.parseInt(theDigits[i]);
    }
    actualLine++;
    line = buffer.readLine();
```

bu kod, txt dosyası içerisindeki tüm elemanları satır satır olacak şekilde theDigits arrayinin içerisine atar. Algoritmayı geliştirirken ana matrisimi integer cinsinden bir array kullanarak devam ettirdim fakat ilerleyen safhalarda integer Array kullanmam, bunları String array'e çevirmem gerektiği için daha fazla satır kod kullanmama sebep oldu. Bu yüzden baştan ana matrisimi String array' e çevirerek işlerime devam ettim.

```
for(int a=0;a<digits.length;a++) {
                for(int b=0;b<digits[0].length;b++) {
                        if(digits[a][b]==0) {
                                M[a][b]="0";
                        }
                        if(digits[a][b]==1)
                        {
                                M[a][b]="1";
                        }
                }
        }
```

Burada her bir integer elemanını String olarak "M" matrisinin içerisine kopyaladım.

```
static int Numberofshapes(String [][] M)
        {

            int ROW = M.length;
            int COL = M[0].length;
            int count = 0;

             for (int i = 0; i < ROW; i++){
                for (int j = 0; j < COL; j++){
                    if (M[i][j] == "1")
                    {
                       count++;
                               if(count==1) {
                            AB1[i][j] = M[i][j];

                             Search1(M, i + 1, j, ROW, COL);     //sağ
                             Search1(M, i - 1, j, ROW, COL);     //sol
                             Search1(M, i, j + 1, ROW, COL);     //yukarı
                             Search1(M, i, j - 1, ROW, COL);     //aşağı
                             Search1(M, i + 1, j + 1, ROW, COL); //sağ üst çarpraz
                             Search1(M, i - 1, j - 1, ROW, COL); //sol alt çarpraz
                             Search1(M, i + 1, j - 1, ROW, COL); //sağ alt çarpraz
                             Search1(M, i - 1, j + 1, ROW, COL); //sol üst çarpraz
                             }
                             if(count==2) {
                                AB2[i][j] = M[i][j];
                             Search2(M, i + 1, j, ROW, COL);     //sağ
                             Search2(M, i - 1, j, ROW, COL);     //sol
                             Search2(M, i, j + 1, ROW, COL);     //yukarı
                             Search2(M, i, j - 1, ROW, COL);     //aşağı
                             Search2(M, i + 1, j + 1, ROW, COL); //sağ üst çarpraz
                             Search2(M, i - 1, j - 1, ROW, COL); //sol alt çarpraz
                             Search2(M, i + 1, j - 1, ROW, COL); //sağ alt çarpraz
                             Search2(M, i - 1, j + 1, ROW, COL); //sol üst çarpraz
                             }

                                        .

                                        .

                                        .
```

En başta matris içerisindeki şekil sayısını bulmayı hedefleyerek başladığım bu fonksiyonu, sonrasında tespit edilen şekilleri bir array içerisinde tutacak şekilde değiştirdim. Bu fonksiyonun eksik olan yönü tespit edilen her bir şekil için fazladan “Search” fonksiyonun kullanılmasının gerekmesiydi. Tek bir fonksiyon içerisinde bunu geliştiremedim. Ödevde teslim edeceğim fonksiyon içerisinde 7 adet search fonksiyonu barındırmakta. Yani bunun anlamı ilk tespit edilen sadece 7ye kadar olan şekillerin yazdırılabilecek olmasıdır. Bulunan şekil sayısı her zaman doğru çalışıyor fakat şekil sayısı search fonksiyonundan fazlaysa hepsini yazdırmıyor. Dilenirse bu search fonksiyonlarının sayısı artırılabilir ve tüm şekiller sorunsuz bir şekilde yazdırılabilir. Çalışma mantığını anlatmam gerekirse;

Ana matrisim olan M matrisi 2 for döngüsünün içerisinde tüm elemanlar taranacak şekilde aranır. Bulunan ilk “1” bizim aslında yeni bir şekil tespit ettiğimiz anlamına gelir. 1 bulunduğu an , içerisinde shape sayısını tutan “count” bir artırılır. Bulunan ilk “1” şeklin kendine özel

olan arrayinin içerisine kopyalanır. Daha sonra şeklin tüm komşu kenarları “Search” fonksiyonuna iletilir.

```
static void Search1(String[][] M, int i, int j, int ROW, int COL)
        {


 if (i < 0 || j < 0 || i > (ROW - 1) || j > (COL - 1) || M[i][j] != "1")
                    {
                        return;
                    }

            if (M[i][j] == "1")
            {
               AB1[i][j] = M[i][j];
                M[i][j] = "0";
                Search1(M, i + 1, j, ROW, COL);     //sağ
                Search1(M, i - 1, j, ROW, COL);     //sol
                Search1(M, i, j + 1, ROW, COL);     //yukarı
                Search1(M, i, j - 1, ROW, COL);     //aşağı
                Search1(M, i + 1, j + 1, ROW, COL); //sağ üst çarpraz
                Search1(M, i - 1, j - 1, ROW, COL); //sol alt çarpraz
                Search1(M, i + 1, j - 1, ROW, COL); //sağ alt çarpraz
                Search1(M, i - 1, j + 1, ROW, COL); //sol üst çarpraz
            }
        }
```

Bu search fonksiyonu recursive bir fonksiyondur tüm komşuları gezene dek kendisini çağırır. Numberofshapeste bulunan 1’in konumu bu fonksiyona iletilir. Sırasıyla bulunan 1’ler şeklin kendisine özel matrisinin içerisine kopyalandıktan sonra bulunan tüm 1’ler sırasıyla sıfırlanır. Kendi içerisinde sürekli döndüğü için baştan sona 1 olan tüm komşuları tarar. Örneğin sol alt çarprazdaki değeri kendisine tekrar gönderir, sonra bu konumun tüm komşularını tarar, eğer burda 1 bulursa kendini tekrar çağırır ve o noktanında tüm komşularını tarar. Ta ki tespit edilen 1 noktalarının komşu noktalarında 1 kalmazsa buradan ayrılır ve Numberofshapes metoduna geri döner. Tespit edilen şeklin tüm 1 değerleri sıfırlandığı için ana fonksiyonda bu noktalar tespit edilmez. Numberofshape metodunda tespit edilen yeni “1” yeni bir şekil olduğunu gösterir ve aynı işlemleri tekrar gerçekleştirir. Bu fonksiyonda en çok zorlandığım kısımlardan birisi matrisin sınırlarında taşma durumu olduğunda fonksiyonun hata vermesiydi. Örneğin -1. Satır olmadığı için program çalışmayı durduruyordu. Bunun için;

```
if (i < 0 || j < 0 || i > (ROW - 1) || j > (COL - 1) || M[i][j] != "1")
                    {
                        return;
                    }
```

Fonksiyonun en başına şöyle bir kod ekledim. Bu if değeri içerisinde bütün illegal durumları barındırmakta. Bunlar olduğu zaman fonksiyon return etmektedir.

Numberofshapes metodunun içerisinde tespit edilen ilk “1” için `ABx[i][j] = M[i][j];`

Bunu kullanmam gereksizdi zaten search fonskiyonlarının içerisinde de bu nokta şeklin arrayine zaten kaydediliyordu. Fakat tek yıldızlı şekiller olduğu zaman bu eşitliği sağlamadığımda, şekli buluyor fakat bunu yazdırmıyordu.

```
public void OutputShapestoFile(String path1) throws IOException {
        //Filewriter kullanarak aynı çıktımı txt dosyama aktarıyorum.
        FileWriter fw = new FileWriter("outputshapes.txt");
        fw.write("There are " + c +" shapes" );
           fw.write("\n");
           for(int aa=1;aa<= c ;aa++) {
                    fw.write("Shape "+ aa);
                    fw.write("\n");
                 for(int i=0;i<11;i++) {
                   for(int j=0;j<14;j++) {
                         if(aa == 1) {
                               fw.write(AB1[i][j]+ " ");
                         }
                         if(aa == 2) {
                               fw.write(AB2[i][j]+ " ");
                               }
                         if(aa == 3) {
                               fw.write(AB3[i][j]+ " ");
                               }
                         if(aa == 4) {
                               fw.write(AB4[i][j]+ " ");
                               }
                         if(aa == 5) {
                               fw.write(AB5[i][j]+ " ");
                               }
                               if(aa == 6) {
                               fw.write(AB6[i][j]+ " ");
                               }
                               if(aa == 7) {
                               fw.write(AB7[i][j]+ " ");
                               }
                   }
                   fw.write("\n ");
                 }
              }
           fw.close();
    }
```

Burada da tespit edilen şekiller dosyanın içerisine yazdırılıyor. Filewriter kullanarak bunu gerçekleştirdim.

Sonuç itibariyle program sorunsuz bir şekilde çalışıyor. Denemelerim sonucunda txt dosyasında yazılacak her türlü şekil doğru şekilde tespit edilip yazdırılıyor. Programın eksik bulduğum yönlerinden birisi tespit edilen her şeklin kendine özel bir matrise ve fonksiyona sahip olması gerekmesiydi. Bu sebepten dolayı defalarca aynı satır kod tekrarlandı fakat sonuç itibariyle istenilen hedefe ulaştım.