

Rozważ zagadnienie perkolacji węzłowej na sieci kwadratowej. Napisz algorytm znajdujący wielkość klastrow na sieci dla ustalonego rozkładu węzłów. (Blok w metodzie init, wywołujący dla każdego węzła metodę klaster_size i drukujący wielkość klastra).

```

import java.awt.*;
import java.applet.*;
// program wyznacza wielkości klastrow-
public class find_clusters extends Applet{
    int siec_size = 8;
    int mnoznik = 30; // wielkość oczka sieci
    int [][] siec = new int [siec_size][siec_size];
    double prob = 0.5; // prawdopodobieństwo obsady
    public void init(){
        setSize(siec_size*mnoznik+10,siec_size*mnoznik+10);
        for (int i=1;i<siec_size-1;i++){
            for (int j=1;j<siec_size-1;j++){

                if(Math.random()<prob) {siec[i][j]= 1;} else
                {siec[i][j]=0;}

            }
        }
        // TODO
    }
    public void paint(Graphics g){
        g.drawLine(1,1,1,siec_size*mnoznik);
        g.drawLine(1,1,siec_size*mnoznik,1);

g.drawLine(siec_size*mnoznik,siec_size*mnoznik,0,siec_size*mnoznik);

g.drawLine(siec_size*mnoznik,siec_size*mnoznik,siec_size*mnoznik,0);
        for (int i=1;i<siec_size-1;i++){
            for (int j=1;j<siec_size-1;j++){

                if(siec[i][j]>0) g.fillRect(i*mnoznik,j*mnoznik,mnoznik,mnoznik);
            }
        }
    }
    int klaster_size(int x, int y){
        if(siec[x][y]==1){
            siec[x][y]=2;
            return 1+klaster_size(x-1,y)+
                klaster_size(x+1,y)+
                klaster_size(x,y-1)+
                klaster_size(x,y+1);
        }
        else
            return 0;
    }
}

```

Oznacz klastry różnymi kolorami. Wykorzystaj następujący konstruktor klasy Color:

`Color`(int rgb)

Creates an opaque sRGB color with the specified combined RGB value consisting of the red component in bits 16-23, the green component in bits 8-15, and the blue component in bits 0-7.

Kolor można ustalić za pomocą metody `setColor`. Np. `g.setColor(new Color(45678345))`

Znajdź największy klaster. Oznacz go losowo wybranym kolorem. Pozostałe klastry zamaluj na jasnoszaro (`Color.LIGHT_GRAY`).

Napisz program na znajdowanie średniej wielkości klastra ($S = \sum l_i^2 / \sum l_i$, gdzie l_i to wielkość i-tego klastra a sumowanie jest po klastrach).

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
public class aver_size_single_run{
    static int siec_size = 300;
    static int [][] siec = new int [siec_size][siec_size];
    public static void main(String[] args){
        double prob = 0.3; // prawdopodobieństwo obsady
        for (int i=1;i<siec_size-1;i++){
            for (int j=1;j<siec_size-1;j++){
                if(Math.random(<prob) {siec[i][j]= 1;}
            }
        }
        for (int i=1;i<siec_size-1;i++){
            for (int j=1;j<siec_size-1;j++){
                int l=klaster_size(i,j);
            }
        }
        try {
            FileWriter outFile = new FileWriter("perkol-test.txt",true);
            PrintWriter out1 = new PrintWriter(outFile);
            out1.println(prob+" "+sumsize2/sumsize);
            out1.close();
        } catch (IOException e){
            e.printStackTrace();
        }
        System.out.println(prob+" "+sumsize2/sumsize);
    } // koniec main
    static int klaster_size(int x, int y){
        if(siec[x][y]!=0){
            siec[x][y]=0;
            return 1+klaster_size(x-1,y)+
                klaster_size(x+1,y)+
                klaster_size(x,y-1)+
                klaster_size(x,y+1);
        }
        else
            return 0;
    }
}
```

Wprowadź uśrednianie po niezależnych próbach (rozkładach węzłów na sieci) oraz wykonaj obliczenia (w pętli) dla `prob=0.2, 0.22, 0.24,...0.6`.

Dokonaj oszacowania punktu krytycznego (wartość dla której średni klaster staje się nieskończenie duży).

W oparciu o otrzymane wyniki wyznacz punkt krytyczny oraz wykładnik γ opisujący rozbieżność średniej wielkości klastra przy zbliżaniu się do punktu krytycznego. Zmodyfikuj algorytm tak aby znajdował klastry na sieci (i) trójkątnej (ii) kubicznej.

Wyznacz dystrybucję klastrów $P(s)$, gdzie s jest wielkością klastra. Jaki zachowuje się $P(s)$ dla dużych s ? Jak zachowanie to zależy od p ?