

Kuis 1

Struktur Data dan Algoritma

IKI20100

**Batas waktu pengumpulan kode sumber:
Selasa, 13 Oktober 2009 pukul 18.15 Waktu Server Ranau**

Kode sumber yang dinilai hanya yang dikumpulkan melalui Ranau. Kode sumber yang dikumpulkan melalui mekanisme selain itu akan diabaikan dan dianggap tidak mengumpulkan.

**Fakultas Ilmu Komputer
Universitas Indonesia**

Sumber Api Pada Kebakaran Hutan

Nama berkas : sda09kuis1.java

Batas waktu eksekusi : 0,5 detik / kasus uji

Batas memori : 8 MB

Latar Belakang

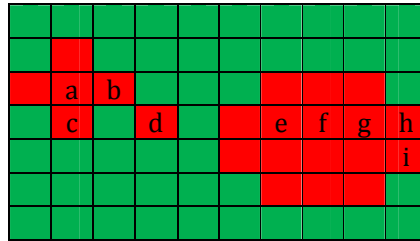
Seperti kita telah ketahui, hutan memiliki banyak fungsi, seperti tempat tinggal dan hidup berbagai satwa, tempat penyimpanan air tanah secara alami, paru-paru dunia, dan lain-lain. Faktanya, selama beberapa dekade terakhir, beberapa hutan di Bumi telah beralih fungsi, telah menyusut, bahkan menghilang. Untuk melindungi hutan-hutan di Bumi, sebuah lembaga internasional bernama Esdea dibentuk. Untuk mendukung pekerjaannya, informasi yang akurat perlu didapat dengan cara yang efisien. Oleh karena itu, Esdea bekerja sama dengan banyak lembaga penelitian internasional.

Saat ini, salah satu proyek yang sedang dikerjakan Esdea adalah mengembangkan sebuah prosedur standar (termasuk berbagai perangkat lunak yang dibutuhkan) untuk menanggulangi dengan cepat jika suatu saat kebakaran hutan sedang terjadi. “Mencegah lebih baik daripada mengobati”, ya kalau bisa mencegah kebakaran, mengapa harus menanggulangnya? Hal ini juga sudah dipikirkan oleh Esdea. Beberapa upaya pencegahan kebakaran telah dilakukan berdasarkan faktor-faktor penyebab kebakaran. Akan tetapi, selalu saja ada faktor X sehingga kebakaran hutan tidak dapat dihindari sama sekali (setidaknya untuk masa sekarang).

Untuk mengetahui apakah terdapat kebakaran pada suatu daerah dengan cepat, mula-mula Esdea perlu mendapatkan citra daerah yang dimaksud. Untuk itu, Esdea memanfaatkan teknologi pencitraan dari satelit-satelit yang mengorbit Bumi. Citra suatu daerah yang direkam dari atas oleh sensor optik satelit merupakan citra yang berbentuk persegi panjang. Citra tersebut dapat dianalogikan sebagai citra yang dihasilkan oleh kamera digital. Bedanya, setiap titik (piksel) pada citra hasil pencitraan satelit mewakili daerah berbentuk persegi seluas beberapa meter persegi. Terkadang citra tersebut tidak dapat menunjukkan gambar daerah sebenarnya yang dimaksud karena terhalang awan atau kabut. Untuk mengatasi hal itu, citra tersebut digabungkan dengan citra lain hasil sensor radar yang dapat menembus awan atau kabut. Citra gabungan itu kemudian disegmentasi menjadi citra tematik berdasarkan hasil penelitian dan pendapat para ahli. Untuk memudahkan, citra tematik merupakan citra yang ukuran dan orientasinya sama, serta terdapat salah satu angka 0 atau 1 yang menempati setiap titik citra tematik yang merupakan hasil pemetaan citra gabungan. Angka 1 menunjukkan pada titik tersebut terdapat api, sedangkan angka 0 sebaliknya.

Sebelum diklasifikasi dan ditetapkan langkah-langkah standar yang harus dilakukan, perlu diketahui terlebih dahulu banyak sumber api dan luas total api pada citra tersebut. Sumber api merupakan kumpulan dari satu atau lebih titik api yang berdekatan. Setiap dua titik api berbeda dikatakan berada pada sumber api yang sama jika kedua titik api tersebut bersebelahan langsung arah 4-tetangga (atas, bawah, kiri, dan kanan) atau

bersebelahan tidak langsung yang dihubungkan oleh satu atau lebih titik api lain (arah 4-tetangga) dari sumber api yang sama. Perhatikan Gambar 1. Titik-titik api a, b, dan c berada pada sumber api yang sama. Titik-titik api e, f, g, h, dan i berada pada sumber api yang sama. Titik api b dan d tidak berada pada sumber api yang sama karena bersebelahan langsung, tetapi bukan arah 4-tetangga. Titik api d dan e tidak berada pada sumber api yang sama karena tidak ada satu titik api pun yang menghubungkan mereka. Titik api e dan i berada pada sumber api yang sama karena keduanya bersebelahan tidak langsung dan dihubungkan oleh titik-titik api f, g, dan h. Luas total api didefinisikan sebagai banyak titik api pada citra tematik.



Gambar 1. Citra Tematik (Diwakili Warna Merah: Api, Hijau: Bukan Api)

Saat ini, Esdea sedang kekurangan tenaga pembuat program sehingga mereka meminta Anda untuk membantu membuat program. Buatlah program yang menerima masukan citra tematik dan dapat menghitung banyak sumber api sekaligus luas total api!

Format Masukan

Masukan dibaca dari standar masukan. Masukan terdiri dari $M+1$ buah baris. Baris pertama berisi dua buah bilangan bulat M dan N ($1 \leq M \leq 80$, $1 \leq N \leq 100$) yang menyatakan ukuran tinggi dan lebar citra tematik dalam satuan titik. M baris berikutnya, masing-masing baris berisi N buah bilangan 0 atau 1 (antar bilangan dipisahkan oleh sebuah spasi).

Format Keluaran

Keluaran ditulis ke standar keluaran. Keluaran berisi dua buah baris. Baris pertama berisi sebuah bilangan bulat yang menyatakan banyak sumber api. Baris kedua berisi sebuah bilangan bulat yang menyatakan luas total api.

Contoh Masukan

```
7 10
0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0
1 1 1 0 0 0 1 1 1 0
0 1 0 1 0 1 1 1 1 1
0 0 0 0 0 1 1 1 1 1
0 0 0 0 0 0 1 1 1 0
0 0 0 0 0 0 0 0 0 0
```

Contoh Keluaran

3 22

Kasus Uji

Terdapat 10 buah kasus uji dengan rincian sebagai berikut.

- Kasus uji 1 berisi masukan yang tidak ada titik api sama sekali.
- Kasus uji 2 berisi masukan yang memiliki sebuah sumber api dan tidak terletak di pinggir citra tematik.
- Kasus uji 3 berisi masukan dengan citra tematik berukuran 1x1 dan titiknya merupakan titik api.
- Kasus uji 4 berisi masukan dengan citra tematik yang berukuran besar dan semua titiknya merupakan titik api.
- Kasus uji 5 sama dengan contoh.
- Kasus uji 6 s.d. 10 menguji kebenaran implementasi program dengan ukuran masukan yang meningkat dan pola sumber api yang bervariasi.

Petunjuk

Gunakan metode rekursif untuk menghitung banyak sumber api. Banyak pernyataan pemanggilan metode rekursif ada empat buah (menyatakan 4-tetangga) di dalam sebuah metode rekursif. Sebelum memanggil metode rekursif, periksalah terlebih dahulu apakah posisi sekarang merupakan batas atas, batas bawah, batas kiri, atau batas kanan citra. Untuk menghindari rekursif tanpa henti, tandai titik-titik yang sudah pernah dikunjungi.

Template Kode Sumber

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.StringTokenizer;

/** Kelas sda09kuis1 ...
    @author Nama (NPM)
 */
public class sda09kuis1
{
    /** Metode main ...
        @param args ...
    */
    public static void main(String[] args)
    {
        // input ...
        BufferedReader input = new BufferedReader(new InputStreamReader(System.in));

        // token ...
        StringTokenizer token;

        try
        {
            token = new StringTokenizer(input.readLine());

            // aThematicImage ...
        }
    }
}
```

```

        ThematicImage aThematicImage = new ThematicImage(
            Integer.parseInt(token.nextToken()),
            Integer.parseInt(token.nextToken()));

        for(int i = 0; i < aThematicImage.getHeight(); i++)
        {
            token = new StringTokenizer(input.readLine());
            for(int j = 0; j < aThematicImage.getWidth(); j++)
            {
                aThematicImage.setCell(i, j,
                    Integer.parseInt(token.nextToken()));
            }
        }

        // Silakan dilengkapi!
    }
    // anException ...
    catch(Exception anException)
    {
        anException.printStackTrace();
    }
}

/** Kelas ThematicImage ...
    @author Nama (NPM)
    */
class ThematicImage
{
    /** height ... */
    private int height;

    /** width ... */
    private int width;

    /** matrix ... */
    private int[][] matrix;

    // Tambahkan field jika diperlukan.

    /** Konstruktor ...
        */
    public ThematicImage()
    {
        this(1, 1);
    }

    /** Konstruktor ...
        @param height ...
        @param width ...
        */
    public ThematicImage(int height, int width)
    {
        setHeight(height);
        setWidth(width);
        matrix = new int[height][width];
    }

    /** Metode setHeight ...
        @param height ...
        */
    public void setHeight(int height)
    {
        this.height = height;
    }
}

```

```

    /** Metode setWidth ...
        @param width ...
    */
    public void setWidth(int width)
    {
        this.width = width;
    }

    /** Metode setCell ...
        @param row ...
        @param column ...
        @param value ...
    */
    public void setCell(int row, int column, int value)
    {
        matrix[row][column] = value;
    }

    /** Metode getHeight ...
        @return ...
    */
    public int getHeight()
    {
        return height;
    }

    /** Metode getWidth ...
        @return ...
    */
    public int getWidth()
    {
        return width;
    }

    /** Metode getMatrix ...
        @return ...
    */
    public int[][] getMatrix()
    {
        return matrix;
    }

    /** Metode getCell ...
        @param row ...
        @param column ...
        @return ...
    */
    public int getCell(int row, int column)
    {
        return matrix[row][column];
    }

    // Tambahkan metode jika diperlukan.
}

// Tambahkan kelas jika diperlukan.

```

Kriteria Penilaian

Terdapat dua bagian penilaian, yaitu:

- nilai kebenaran program sebesar 70%.
- nilai dokumentasi dan gaya pemrograman sebesar 30% (detail lihat di SCell, <http://scele.cs.ui.ac.id/s1/mod/resource/view.php?id=16765>).