IPAR 4.0 .NET C# ALAPOKON

6. forduló

SEMILAB

A kategória támogatója: Semilab Zrt.

Ismertető a feladathoz

A feladatlap több csatolmányt is tartalmaz, ezért a megoldását asztali gépen javasoljuk!

Fontos!

Fordulónként javasoljuk az összes részfeladat végigolvasását a kidolgozás megkezdése előtt, mivel a feladatok sokszor egymásra épülnek. Előfordul, hogy egy részfeladat nehézségét az input mérete adja, így érdemes hatékony megoldásokra törekedni.

Felhasznált idő: 00:00/40:00 **Elért pontszám:** 0/15

Indítás előtti csatolmányok

1. feladat 0/3 pont

Egy mérőgépünk segítségével négyzetes wafereket mértünk le, egyes wafereket akár többször is. A mérés eredménye egy NxN mátrix, ahol a mátrix pontjai a wafer adott pontban mért magasságát adják meg. Egy wafer ismételt lemérése során mindig pontosan ugyanazokat a magasságokat kapjuk. Az inputfájlunk tartalmazza a waferek méretét egy integerként (N), illetve a mérési eredményeket egymás után (k db NxN mátrix formájában). Az inputunk alakja a következő:

Első sor: N – a waferek mérete

További sorok [wafer darabszám]*N db sor: N hosszú 0-9 értékekből álló karaktersorozatok

Adjuk meg a 6_1.txt alapján, hogy az ott szereplő első wafert összesen hányszor mértük le (az első mérést is beleszámolva)!

Példa:

(3 db 2x2es eredmény az inputon)

Input:

2

01

03

05

69

01

03

Output:

2

A megoldás 2. Az első waferhez tartozó eredmény [[0,1],[0,3]], ami kétszer szerepel az inputon.

A 6_1_test.txt eredménye 5.

Segítség a beolvasáshoz (nem kötelező használni):

```
static List<List<List<int>>> ReadInput(string inputFile)
{
    var fileLines = File.ReadAllLines(inputFile);
    var N = int.Parse(fileLines[0]);
    var wafers = File.ReadLines(inputFile)
    .Skip(1)
    .Select(s => s.ToCharArray()
        .Select(c => (int)c - '0')
            .ToList())
    .Select((v, i) => new { Index = i, Value = v })
    .GroupBy(k => k.Index / N)
    .Select(g => g
        .ToList()
            .Select(v => v.Value)
                .ToList())
    .ToList();
    // wafers[i][y][x]: i - edik wafer y, x pixelpozícióban
    return wafers;
}
```

Válasz

A helyes válasz:

4

Magyarázat

```
public static int Solve(string inputFile)
{
        var wafers = ReadInput(inputFile);
        var reference = wafers[0];
        int sum = 0;
        foreach (var wafer in wafers)
        {
                bool isReference = true;
                for (int y = 0; y < reference.Count && isReference; y++)
                {
                        for (int x = 0; x < reference[y].Count && isReference; x++)
                        {
                                if (reference[y][x] != wafer[y][x])
                                {
                                        isReference = false;
                                }
                        }
                }
```

```
                if (isReference)
                {
                        sum++;
                }
        }
        Console.WriteLine(sum);
        return sum;
}

static List<List<List<int>>> ReadInput(string inputFile)
{
        var fileLines = File.ReadAllLines(inputFile);
        var N = int.Parse(fileLines[0]);
        var wafers = File.ReadLines(inputFile)
        .Skip(1)
        .Select(s => s.ToCharArray()
                .Select(c => (int)c - '0')
                        .ToList())
        .Select((v, i) => new { Index = i, Value = v })
        .GroupBy(k => k.Index / N)
        .Select(g => g
                .ToList()
                        .Select(v => v.Value)
                                .ToList())
        .ToList();
        return wafers;
}
```

2. feladat 0/6 pont

Egy olyan programot készítünk, ami megszámolja, hogy összesen hány különböző wafert mértünk le. A programunkat fel kel készíteni arra, hogy 0, 90, 180 vagy 270 fokkal elforgatva is rakhatnak be wafereket a mérőgépbe. Fel kell tudni ismerni, hogy annak ellenére, hogy egy wafert a megadott fokokkal elforgatva raknak a gépbe, még ugyanannak a wafernek számít a különbözőség vizsgálatakor.

Adjuk meg, hogy hány különböző wafert mértünk le a fenti feltételek mellett a 6_2.txt-ben!

Példa:

Input:

2

01

03

05

69

30

10

00

31

01

03

Output:

2

Ebben az esetben 2 különböző waferünk van, mert az utolsó három az az elsőnek az elforgatottjai.

A 6_2_test.txt inputra a helyes megoldás 115.

Válasz

A helyes válasz:

109

Magyarázat

Algoritmus leírása:

Minden waferhez hozzárendelünk egy string hash-t. Az hash bármi lehet, ami képes egyedileg azonosítani a wafert, mi a mátrix elemeinek sorfolytonos kiolvasásából aggregált sztringet választottuk.

Végigmegyünk az összes waferen. Legyen egy sum változónk, amely tartalmazza az összes különböző wafer elemszámát az aktuálisan vizsgált elemig. Továbbá legyen a hashsetünk, amelybe a wafer hasheket tudjuk pakolni.

Minden wafernél a következő műveleteket végezzük el:

- Megnézzük, hogy a wafer 0,90,180,270 elforgatásainak a hash-e szerepel-e a hashsetünkben.
- Ha szerepel, akkor megyünk tovább
- Ha nem szerepel, akkor a sum változónkat növeljük 1-el, és mind a 4 elforgatás hash-ét belerakjuk a hashstbe.

A végén a sum változó fogja tartalmazni a különböző wafereink számát.

```
public class Wafer
{
    public List<List<int>> Matrix { get; private set; }
    public string WaferHash { get; set; }
    public Wafer(List<List<int>> matrix)
    {
        Matrix = matrix;
        WaferHash = string.Join("", matrix.SelectMany(i => i));
    }


    public Wafer Create90Rotated()
    {
        //source : https://www.csharp-console-examples.com/loop/for-loop/rotate-matrix-to-90-degree-in-c/
        var N = Matrix.Count;
        List<List<int>> rotatedWafer = new int[N]
            .Select(e => new int[N]
                .ToList())
            .ToList();
        int j = 0;
        int p = 0;
        int q = 0;
        int i = N - 1;

        for (int k = 0; k < N; k++)
        {
            while (i >= 0)
            {
                rotatedWafer[p][q] = Matrix[i][j];
                q++;
                i--;
            }
            j++;
            i = N - 1;
```

```
            q = 0;
            p++;
        }
        return new Wafer(rotatedWafer);
    }
}

static List<List<List<int>>> ReadInput(string inputFile)
{
    var fileLines = File.ReadAllLines(inputFile);
    var N = int.Parse(fileLines[0]);
    var wafers = File.ReadLines(inputFile)
    .Skip(1)
    .Select(s => s.ToCharArray()
        .Select(c => (int)c - '0')
            .ToList())
    .Select((v, i) => new { Index = i, Value = v })
    .GroupBy(k => k.Index / N)
    .Select(g => g
        .ToList()
            .Select(v => v.Value)
                .ToList())
    .ToList();
    return wafers;
}


public static int Solve(string inputFile)
{
    var wafers = ReadInput(inputFile).Select(w => new Wafer(w));
    var allWaferCode = new HashSet<string>();
    int sum = 0;
    foreach (var wafer in wafers)
    {
        if (allWaferCode.Contains(wafer.WaferHash)) continue;
        sum++;
        var rotated = wafer;
        for (int i = 0; i < 4; i++)
        {
            allWaferCode.Add(rotated.WaferHash);
            rotated = rotated.Create90Rotated();
        }
    }
    return sum;
}
```

3. feladat 0/6 pont

Adjuk meg, hogy hány db különböző wafert mértünk le a 6_3.txt-ben, a 2. feladatban megadott forgatási szabályok érvényessége mellett!

Válasz

A helyes válasz:

99125

Magyarázat

A 2. feladatban megadott megoldás erre az inputra is belátható időn belül jó eredményt ad.