

IPAR 4.0 .NET C# ALAPOKON

5. forduló



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: 39:10/40:00

Elért pontszám: 2/12

Indítás előtti csatolmányok

1. feladat 2/2 pont

Mérőberendezésünkkel wafereket vizsgálunk, melyekhez két tulajdonságot rendelünk: egy típust (A,B,C ... G) és egy minőségi osztályt (1-10). **A megrendelő kérésére egy olyan adott átlagminőségű wafercsomagot szeretnénk összeállítani, amiben minden típusú waferből pontosan egy darab szerepel.**

Egy csomag átlagminősége megegyezik a benne szereplő wafererek minőségének átlagával.

Két csomagot akkor tekintünk egyezőnek, ha pontosan ugyanazok a waferek alkotják. Az inputban szereplő waferek különbözőek, még akkor is, ha tulajdonságaikban megegyeznek.

Az input formája:

ExpectedAvgQuality

waferType_1 waferType_2 ... waferType_n

quality_1 quality_2 ... quality_n

Az elvárt átlagminőség (ExpectedAvgQuality) egész szám, minden wafertípus (waferType_i) A és G közötti nagybetűs karakter, a minőségek (quality_i) pedig egész számok 1 és 10 között.

Példa:

Input:

6

A B B A C C A

5 10 7 6 2 5 5

Output:

2

A válasz 2, hiszen két ilyen csomag létezik:

(A, 6) (B, 10) (C, 2) hiszen $(6 + 10 + 2) / 3 = 6$

(A, 6) (B, 7) (C, 5) hiszen $(6 + 7 + 5) / 3 = 6$

Hányféleképpen tudunk összeállítani a feltételeknek megfelelő csomagot az 5_1.txt-re?

Az 5_1_test.txt-re adott helyes eredmény 5.

Válasz

8

A helyes válasz:

8

Magyarázat

```
public static class WaferPackageQuality
{
    public static long Solve(string fileName)
    {
        var lines = File.ReadAllLines(fileName);
        var expectedAvgQuality = long.Parse(lines[0]);
        var waferTypes = lines[1].Split(' ').ToArray();
        var qualities = lines[2].Split(' ').Select(int.Parse).ToArray();
        var waferQualities = waferTypes.Select((wt, i) => new WaferQuality(wt, qualities[i])).ToArray();

        return GetNumberOfPackageCombinations(expectedAvgQuality, waferQualities);
    }

    static long GetNumberOfPackageCombinations(long expectedAvgQuality, IEnumerable<WaferQuality> waferQualities)
    {
        // összevonjuk az azonos (típus, minőség) tulajdonságú wafereket és típusonként külön listába
        var qualitiesByType = waferQualities
            .GroupBy(a => new { a.WaferType, a.Quality })
            .Select(b => new WaferQuality(b.Key.WaferType, b.Key.Quality, b.Count()))
            .GroupBy(c => c.WaferType)
            .Select(d => d.Select(e => e.Quality).Zip(d.Select(f => f.Count), (q, count) => new { q, count }));

        // előállítjuk a minőségi csoportok összes lehetséges kombinációját és ha a minőségek átlaga
        // minőség-kombinációhoz tartozó összes wafer-kombináció számát
        long totalCount = 0;
        foreach (var combination in GetCombinations(qualitiesByType))
        {
            if (combination.Sum(c => c.Quality) == expectedAvgQuality * combination.Length)
            {
                totalCount += combination.Aggregate((long)1, (a, b) => a * b.Count);
            }
        }

        return totalCount;
    }
}
```

```

static List<T[]> GetCombinations<T>(params IEnumerable<T>[] enumerables)
{
    var combinations = new List<T[]>(enumerables.Length);
    if (enumerables.Length > 0)
    {
        var chain = new T[enumerables.Length];
        GetCombinationsRec(enumerables, chain, 0, combinations);
    }
    return combinations;
}

static void GetCombinationsRec<T>(IList<IEnumerable<T>> sources, T[] chain, int index, ICollection<T[]> combinations)
{
    foreach (var element in sources[index])
    {
        chain[index] = element;
        if (index == sources.Count - 1)
        {
            var finalChain = new T[chain.Length];
            chain.CopyTo(finalChain, 0);
            combinations.Add(finalChain);
        }
        else
        {
            GetCombinationsRec(sources, chain, index + 1, combinations);
        }
    }
}

public class WaferQuality
{
    public string WaferType { get; set; }
    public int Quality { get; set; }
    public long Count { get; set; }

    public WaferQuality(string waferType, int quality, long count = 1)
    {
        WaferType = waferType;
        Quality = quality;
        Count = count;
    }
}

```

2. feladat 0/3 pont

Az 1. feladat feltételei alapján hányféleképpen tudunk összeállítani a feltételeknek megfelelő csomagot az 5_2.txt-re?

Az 5_2_test.txt-re adott helyes eredmény 591.

Válasz

A helyes válasz:

632

Magyarázat

A 1. feladatban megadott megoldás erre az inputra is belátható időn belül jó eredményt ad.

3. feladat 0/7 pont

Az 1. feladat feltételei alapján hányféleképpen tudunk összeállítani a feltételeknek megfelelő csomagot az 5_3.txt-re?

Az 5_3_test.txt-re adott helyes eredmény 55139785963616.

Válasz

A helyes válasz:

55582308328615

Magyarázat

A 1. feladatban megadott megoldás erre az inputra is belátható időn belül jó eredményt ad.



[Legfontosabb tudnivalók](#) [Kapcsolat](#) [Versenyszabályzat](#) [Adatvédelem](#)

© 2023 Human Priority Kft.

KÉSZÍTETTE **cone**

Megjelenés

Világos