Bővelkedő lottószámok:

1. [Specifikáció:](https://progalap.elte.hu/specifikacio/?data=H4sIAAAAAAAACm2QTWrDMBCFrzJoZYPsxok3HUjAhbaUJO6iTRZ1vJAjGQSRWmylxA1ZdJd79SY9SSX%2FpFC60Xx6M%2B9p0JHUb2IrS7llRr5qguRGIKjv8zmloLuiGGeVwywKQ0XBnjrfVBs9lwi86IaaYYAXbe%2FuHaH%2B2CtMg5ltO%2FQOPkzh6WW1XCaenLqYPjw7UJn7g60UO%2BNsC9piZ1sn6T%2Beqz53NgqvW%2F9tiRA4WFnweEEbZ54%2FrL8%2BF4%2F3yXOXofpk6VOQPqHEiNrUBLMj4cww%2BwsKIXYxGmHiavskOgIIIIsojCjE%2BUUYt0L0K9ib0%2FI%2Flk7gBcLYQXOJnAwQ23U0U8IuYakS9X5nCEan%2FPQDsX4SmqwBAAA%3D)

Be: m∈N, n∈N, madar∈N[1..m, 1..n]

Ki: db∈N, y∈N[1..db]

Fv: szum:N->N, szum(x) = SZUMMA(i=1..n, madar[x,i])

Fv: felt:N->L, felt(x) = VAN(i=1..n, madar[x,i]/szum(x)>0.9)

Ef: -

Uf: (db,y) = KIVÁLOGAT(i=1..m, felt(i), i)

2. Visszavezetés:

Kiválogatás: Eldöntés: Összegzés:

e..u ~ 1..m e..u ~ 1..n e..u ~ 1..n

T(i) ~ felt(i) T(i) ~ madar[x,i]/szum(x)> 0.9 f(i) ~ madar[x,i]

f(i) ~ i

A diagram of a line

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated3. Algoritmus:

namespace helysegTobbMint90

{

internal class Program

{

static void Main(string[] args)

{

int[,] madar = beolvasas();

(int db, List<int> y) = feldolgozas(madar);

kiiras(db, y);

}

static int[,] beolvasas()

{

int[,] madar;

int m, n;

Console.Error.Write("m n = ");

string[] be1 = Console.ReadLine()!.Split(" ");

int.TryParse(be1[0], out m);

int.TryParse(be1[1], out n);

madar = new int[m, n];

for (int i = 0; i < m; i++)

{

Console.Error.Write("{0}. szoba madarai: ", i+1);

string[] be2 = Console.ReadLine()!.Split(" ");

for (int j = 0; j < n; j++)

{

int.TryParse(be2[j], out madar[i, j]);

}

}

return madar;

}

static (int db, List<int> y) feldolgozas(int[,] madar)

{

int db; List<int> y;

db = 0;

y = new List<int>();

bool felt(int x)

{

bool van = false;

int i = 0;

while (i < madar.GetLength(1) && !((double)madar[x, i] / szum(x) > 0.9)) i++;

van = i < madar.GetLength(1);

return van;

}

int szum (int x)

{

int s = 0;

for (int i = 0; i < madar.GetLength(1); i++)

{

s += madar[x, i];

}

return s;

}

for (int i = 0; i < madar.GetLength(0); i++)

{

if (felt(i))

{

db++;

y.Add(i+1); //0-tól való indexelés miatt a helység száma az indexe + 1

}

}

return (db, y);

}

static void kiiras(int db, List<int> y)

{

Console.Error.Write("db = ");

Console.WriteLine(db);

Console.Error.Write("ezek a helységek: ");

for (int i = 0; i < db; i++)

{

Console.Write(y[i] + " ");

}

}

}

}