**Cellár László: Bűvös csiga**

A beadandó dokumentációja

1. fázis: leírás és állapottér

Egy egyszemélyes játék, a Bűvös csiga pályaszerkesztőjét és megoldóját valósítom meg.

**Játékleírás:**

Adott egy n\*n mezőből álló négyzet alakú tábla, amelynek egyes mezőiben 1 és m közötti számok szerepelnek.

A feladat az, hogy további 1 és m közötti számokat helyezzünk el a táblában úgy, hogy az alábbi feltételek teljesüljenek:

1. Minden sorban és minden oszlopban az 1..m számok mindegyike pontosan egyszer szerepel.
2. A bal felső sarokból induló csigavonal mentén (melyet a nyilak jelölnek) a számok rendre az 1,2,...m,1,2,...,m,... sorrendben követik egymást. (A csigavonal utolsó pontján nem található iránymutatás, ez jelzi a pálya végét.)

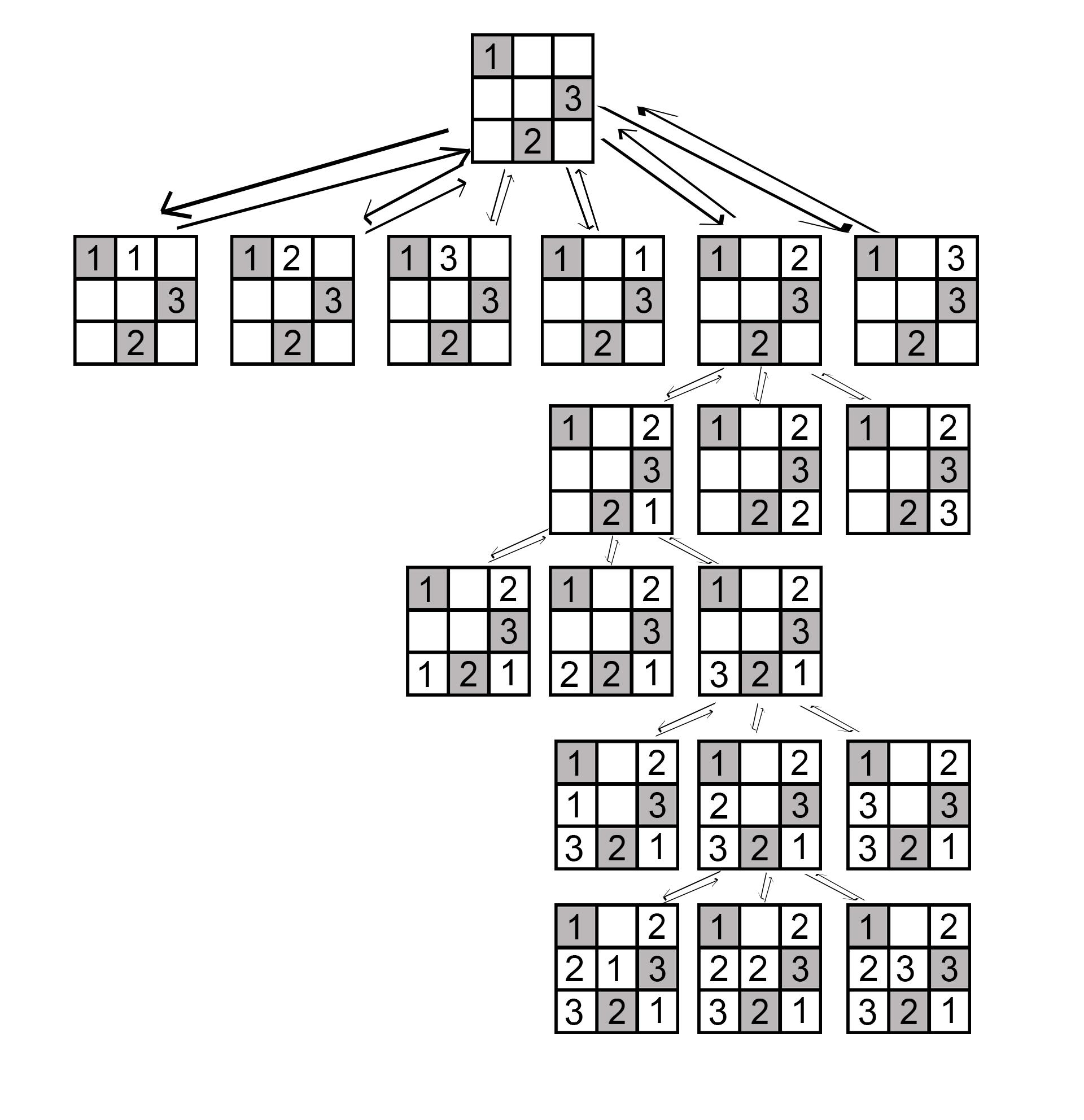
**Kezdőállapot:** A pályaszerkesztés eredménye.

A megszerkesztett pályát a játékban egy kétdimenziós int tömbben tárolom, a tömb mérete a pálya méretétől függ. A kétdimenziós int tömbből a program készít egy egydimenziós int tömböt, hogy könnyebb legyen megállapítani, hogy a számok folytonosan követik-e egymást vagy sem.

**Lehetséges lépések:** Bármely mező értékének növelése egy tetszőleges értékre 1 és m között.

**Állapottér-reprezentáció példa egy helyes megoldásra:** Vegyünk egy 3x3-as tömböt, melyben a számokat 1…3-ig kell elhelyezni. A pályaszerkesztőben beírt számokat a játék során nem lehet módosítani, ezt szürkével jelölöm.

Továbbá a játék során nem követelmény, hogy minden mező ki legyen töltve.

****

A kiindulási tömb így néz ki:

[1,0,0]

[0,0,3]

[0,2,0]

Ebből készít a program egy körkörös tömböt, mely így néz ki:

1, 0, 0, 3, 0, 2, 0, 0, 0

Ebben a programban nincs helyesebb, gyorsabb, jobb megoldás, a cél csak egy helyes megoldás kirakása, így minden lépés költsége 1.

A Luger keretrendszernek a State interfészét az AbstractState osztályton keresztül valósítom meg.

Az palyaFelepitese osztálynak több dinamikus adattagja van:

* int[][] palya: a pálya felépítése tömbként
* int PalyaMerete: a pálya mérete
* int SzamokMeddig: megadható maximális szám
* int[] korkorosSzamok: a pálya körkörös felépítése
* String[][] korkorosIranyok: a haladási irányt mutatja

**Megvalósított algoritmus:** Best First algoritmus