Hanoi tornyai

Innen: Algowiki

Tartalomjegyzék

- 1 A feladat
- 2 Megfigyelések
- 3 Lassú megoldás
- 4 Ötletek
- 5 Megoldás
- 6 Komplexitás
- 7 Implementáció

A feladat

Tekintsük a jól ismert Hanoi tornyai problémának azt a változatát, amikor a kezdeti játékállásban és a cél játékállásban is a korongok tetszőlegesen helyezkedhetnek el, feltéve, hogy mindegyik nála nagyobb korongon van (vagy az alsó). A játék során egy lépésben egy korongot mozgathatunk valamelyik torony tetejéről egy másik torony tetejére, ha ott nálánál nagyobb korong van. A feladat az, hogy rakjuk át egyesével mozgatva a korongokat, betartva, hogy korongot csak nála nagyobbra rakhatunk.

Készítsünk programot, amely megad egy olyan lépéssorozatot, amely hatására a kezdeti játékállásból a cél játékállás keletkezik!

Fontos, a legnagyobb korong mérete 16 lehet!

Ha egy korong n méretű, akkor minden n-nél kisebb korong biztosan szerepel a korongok között.

(Mester, Haladó: Rekurzív kiszámítás, 14 Hanoi tornyai variáns)

Megfigyelések

Mivel 3 pozíciónk van, és mindhárom helyen különböző nagyságú korongok vannak*, világos hogy a korongok között lesz egy legkisebb, egy legnagyobb és egy középső. *(Ha az egyik helyen nincs korong, akkor nyilván az a legnagyobb, lehet viszont minden korong egy helyen és akkor két üres helyünk marad.)

A lépések amiket tehetünk: a legkisebb korongot mindkét másikra át lehet tenni. A legnagyobb korongot nem tudjuk mozgatni. A közepes méretű korongot a nagyra át tudjuk tenni. Ahhoz , hogy a legnagyobbat a három közül majd mozgathassuk, valamelyik rúdon helyet kelle neki csinálni, azaz minden nála kisebbet át kell mozgatni a 3. helyre.

Minden korong pontosan egy rúdon lehet, és egy rúdon a rajta lévők között egyértelmű a sorrend. Összesen maximum 3^16 féle játékállás lehetséges.

Lassú megoldás

Minden helyzetnél vizsgáljuk a lehetséges további lehetőségeket, azaz az összes lehetséges mozgatást végezzük el. A lépéseket is számon kell tartanunk, továbbá hogy ne próbálgassunk oda-vissza tenni egy korongot, valahogy vizsgálnunk kell, hogy egy adott pozíciót vizsgáltunk-e már. Ezt meg tudjuk tenni akár egy n elemű (n 16) tömbbel.

Ily módon minden mozgatást ki fogunk próbálni, és biztosan helyes megoldást fogunk kapni, azonban a 3^{16} pozíció és még a vizsgálatok túl lassúak lesznek.

Ötletek

Közelítsük a másik irányból a problémát!

- A legalsó korongot biztosan elég 1-szer mozgatni. Ahhoz hogy jó pozícióba kerüljön, a mozgatásakor az új helyén semmi sem lehet, azaz üresnek kell lennie.
- Ahhoz, hogy mozgatni tudjuk a legalsó korongot, mindent ami a kiindulási rúdján felette van és mindent, ami a cél rúdon van, át kell mozgatnunk a nem használt rúdra.
- Ha a legalsó korong már jó helyen van, akkor akár úgy is tehetünk, mintha nem létezne, hiszen nála minden kisebb, mindent rá lehet rakni.
- A klasszikus Hanoi megoldást használva tudunk módszert adott korongok adott rúdra mozgatására. https://hu.wikipedia.org/wiki/Hanoi_tornyai

Megoldás

A legnagyobb korongtól kezdve mozgassunk mindent a helyére, a klasszikus Hanoi megoldást használva. Az aktuális mozgatandó korongnál minden kisebbet helyezzünk a 3. rúdra, rakjuk az aktuális korongot a cél rúdjára! Ez után, az utána következő kisebb koronggal végezzük el ugyanezt, ameddig a legkisebb korong a helyére nem kerül!

Komplexitás

Tudjuk, hogy a sima átrendezéshez 2^n lépés szükséges. A legrosszabb esetben ezt kell megtennünk minden korong esetén, azaz n alkalommal. Ebben az esetben a futási idő komplexitása $O(n * 2^n)$.

A helyek eltárolásához az aktuális- (kezdetben a beolvasott pozíciók) és végpoziók 1-1 n elemű tömbje elegendő.

Implementáció

Az implementáció c++ nyelven: https://pastebin.com/gq93a9fi