The Execution Time of N Figures Problem Cannot Be Polynomial

Written by Jozsef Kiss (KissCode Systems Kft, Hungary)
Site: http://kcsopensource.com/
Date: 01.01.2018.

0. ABSTRACT

Letezik az algoritmusoknak azon nemures reszhalmaza, amely elemei az N sakkfigura (N kiralyno es kiterjesztese) problemat megoldjak. Ezen algoritmusok implementacioira tapasztalat szerint jellemzo, hogy futasi idejuk drasztikusan megnovekszik ahogy a Sakktabla meretet (es a lerakni kivant sakkfigurak szamat) noveljuk.
A jelenlegi vilagrekord vezerre 27 meretu tablan van kiszamolva, ezen a Sakktablan ismeretes tehat a vezerek osszes olyan lerakasa amelyekben egyik vezer sem tamadja a masikat.
A Futasi ido roviditesere jo sok otlet gyult mar ossze (bitenkenti muveletek hasznalata, szimmetria hasznalata, tamadas elokalkulacio + csak ervenyes poziciok hasznalata, stb.) azonban a tendencia nem tudott szamottevoen valtozni. Ha tudott volna, akkor ismernenk az osszes helyes kiralyno lerakast mondjuk 100 meretu tablan, de nem ismerjuk. Az alabbiakban szeretnenk megmutatni azt, hogy az N sakkfigura osszes lehetseges helyes lerakasa es ezek megszamolasa nem lehetseges gyorsan azaz Polynomialis futasi idoben.
Az N kiralyno problema gyors megoldasaval egyebkent cafolni lehetne a P!=NP sejtest amely egyike a milleniumi problemaknak.

1. DEFINITIONS

```
1.1  Dimenzio ( D )
     osszes figura darabszam amelyet egyszerre hasznalni szeretnenk
1.2  Sakktabla ( C )
     0 .. ( D ^ 2 - 1 ) rendezett sorozat
     peldaul a C4x4-es Sakktabla reprezentacioja: 0  1  2  3
                                                  4  5  6  7
                                                  8  9  10 11
                                                  12 13 14 15
1.3  Jo pozicio
     C tetszoleges 2 eleme: Ci es Cj ( i,j eleme N es i,j <= D ^ 2 - 1 )
     kozott ertelmezett, es azon tulajdonsag amely szerint az emlitett 2
     sakkfigura nem tamadja egymast
1.4  Jo lerakas
     C Sakktabla D darab eleme amelyek kozul mindegyik 2 elem Jo pozicio
1.5  N sakkfigura problema
     minden Jo lerakast eloallitani a D szamu sakkfigurakra a D meretu C
     Sakktablan, es megszamolni ezeket
1.6  N sakkfigura algoritmus ( A )
     olyan algoritmus amely az osszes sakkfigura lerakasat elvegzi az osszes
     helyes lerakasban optimalis modon, azaz nem tartalmaz olyan kalkulaciot
     amely nem az N sakkfigura problema megoldasan dolgozna
1.7  Ellenorzesi ido ( T0 )
     az az idotartam amely alatt a szamitogep ellenorzi ket tetszoleges Ci, Cj
     elemekrol hogy azok Jo pozicioban vannak-e vagy sem
     nem fugg a pillanatnyi idoponttol, sem pedig az osszehasonlitott 2
     elemtol
     kicsiny konstanskent kezeljuk
1.8  Megszamolasi ido ( T1 )
     az az idotartam amely alatt a szamitogep egy helyes lerakast megszamol
     T0-hoz hasonloan kicsiny konstanskent kezeljuk
1.9  Futasi ido ( TR )
     az a teljes idotartam amely ido alatt A implementacioja elkezdi
     majd be is fejezi a futasat, azaz a kezdo es vegzo idopillanatok
     kulonbsege
     TR csakis attol fugg hogy hany elemi CPU utasitast kell vegrehajtani
     igy kozvetett modon D fuggvenye
1.10 Polynomialis futasi ido
     letezik k pozitiv egesz szam hogy az A algoritmus implementacio Futasi
     ideje O ( D ^ k ) osszefugges szerint alakuljon tetszoleges D ertekre
```

2. ANALYSIS

2.1 Assumption: lehetseges Polynomialis futasi ido alatt megoldani az N sakkfigura problemat az A algoritmus implementaciojanak futtatasaval.

2.2 Az N sakkfigura problema alapveto elvarasa hogy A:
- kepes legyen eldonteni C barmely reszsorozatarol hogy azok elemei Jo poziciok vagy sem
- a Jo lerakasokat kepes legyen megszamolni

2.3 Az A algoritmustol pontosan a fontieket koveteljuk meg es nem tobbet.
Tehat, A legyen olyan tulajdonsagu hogy D ertekebol egyertelmuen es pontosan 0 ido alatt bocsassa rendelkezesunkre az osszes helyes lerakast alkoto reszsorozatokat, rossz lerakassal ne talalkozzunk.
Tovabba, A kepes legyen eldonteni ezekrol a C reszsorozatokrol hogy minden 2 eleme Jo pozicio legyen, es a Jo lerakasokat kepes legyen osszeszamolni.

2.4 Az A algoritmus tehat a kovetkezo feladatokat vegzi el:

D darab elemrol

$$0 + 1 + 2 + \dots + (D - 1) = 1/2 * D * (D - 1) \qquad (1)$$

darab ellenorzessel tudja meghatarozni hogy C adott reszsorozata Jo lerakas.
Ez (1) alapjan

$$T0 * 1/2 * D * (D - 1) \qquad (2)$$

ideig tart.
Ezt az uj poziciot hozza kell adni a mar megtalalt darabszamhoz (found ++), ami T1 ideig tart.
Legyen

$$G = G(D) \qquad (3.a)$$

az a darabszam amely a D Dimenziohoz tartozo helyes sakkfigura lerakasok szamat jeloli.
Nem tudunk rola semmit csak annyit hogy nagy D ertekek eseten

$$G(D) < G(D + 1) \qquad (3.b)$$

Ekkor (1), (2) es (3.a) alapjan a fontiek elvegzesehez szukseges osszes idotartam:

$$TR(D) = T0 * 1/2 * D * (D - 1) * G + T1 * G \qquad (4)$$

A tovabbiakban nagy D ertekeket vizsgalunk, igy

$$D * (D - 1) \sim D^2 \qquad (5)$$

Azaz a (4) osszefugges igy alakul nagy D ertekekre:

$$TR(D) = T0 * 1/2 * D^2 * G + T1 * G \qquad (6)$$

Legyen 1.10 es (6) alapjan:

$$\begin{aligned} TR(D) &= D^k \\ T0 * 1/2 * D^2 * G + T1 * G &= D^k \end{aligned} \qquad (7)$$

azaz, keressunk allando k erteket ennek a bal oldalon levo Futasi idonek.

2.5 Nezzuk meg hogy a 2.4 pont (7) osszefuggesere milyen k ertekeket lehet talalni.

2.5.1 Legyen k = 1, ekkor (7):

$$\begin{aligned} T0 * 1/2 * D^2 * G + T1 * G &= D \\ 1/2 * T0 * G * D^2 - D + T1 * G &= 0 \end{aligned} \qquad (8)$$

Ez D-re egy masodfoku egyenlet, amelynek akkor van legalabb 1 megoldasa, ha

```
  ( 9 )     ( - 1 ) ^ 2 - 4 * 1/2 * T0 * G * T1 * G >= 0
                            1 - 2 * T0 * T1 * G ^ 2 >= 0
```

T0 es T1 kicsiny konstansok, viszont (3.a) miatt G fuggvenye az aktualis D Sakktabla meretnek. Igy egy bizonyos D ertek folott (3.b) miatt (9) nem teljesulhet.
Ezert (7) nem teljesulhet minden egyes D esetben k = 1-re.

2.5.2 Legyen k = 2, ekkor (7):

```
  ( 10 )           T0 * 1/2 * D ^ 2 * G + T1 * G = D ^ 2
            ( 1/2 * T0 * G - 1 ) * D ^ 2 + T1 * G = 0
                     ( 1 - 1/2 * T0 * G ) * D ^ 2 = T1 * G
                                                D = sqrt (
                                                 T1 * G / ( 1 - 1/2 * T0 * G )
                                                         )
```

Ez akkor ertelmezett, ha

```
  ( 11 )    1 - 1/2 * T0 * G > 0
```

A (9)-hez hasonloan (11)-ben is az latszik hogy (3.b) miatt nem teljesulhet mindig, ezert (7) sem teljesulhet tetszoleges D ertekre k = 2 eseten.

2.5.3 Vizsgaljuk (7)-et altalanos azaz valtozatlan formajaban.

```
  ( 12 )                      T0 * 1/2 * D ^ 2 * G + T1 * G = D ^ k , k > 2
                        D ^ k - 1/2 * T0 * G * D ^ 2 - T1 * G = 0
            D ^ 2 * ( D ^ ( k - 2 ) - 1/2 * T0 * G ) - T1 * G = 0
```

Mivel T1 * G > 0 es D ^ 2 > 0 ezert a fontebbi csak akkor teljesulhet, ha

```
  ( 13 )    D ^ ( k - 2 ) - 1/2 * T0 * G > 0
```

Ez azt jelenti, hogy adott D-hez mindig lehet legkisebb k erteket talalni, de k nem lehet allando: ha D novekszik akkor k-nak is novekednie kell.
Ez azert van mert G (D) gyorsabban novekszik mint D ^ (k - 2):
- vezer eseten G (D) felulbecsulheto: D! / (2 ^ D) szerint
- bastya eseten G (D) pontosan D!
- futo eseten G (D) az D! relacional is meredekebb.

2.5.4 Igy k = 1, k = 2 es altalanos k esetre is lathato hogy (7) nem teljesulhet tetszoleges D esetre. Ennel fogva 2.1 Assumption sem igaz.

2.6 A fonti megfontolasok soran nem szamoltunk azzal hogy a jo es rossz lerakasok megkeresesehez is szukseges 0-nal tobb ido. Ez a (6) osszegben egy tovabbi tagkent jelenik meg, amely fuggvenye a D-nek, es legyen

```
  ( 14.a )  TS = TS ( D ) != 0
```

A (3.b)-bol kovetkezik hogy nagy D-kre

```
  ( 14.b )  TS ( D ) < TS ( D + 1 )
```

tovabba 2.5.3-bol lathato TS (D) tendenciaja (ami legalabb akkora mint G (D) meredeksege mert jo es nem jo lerakasok mindig lesznek es egyre tobb)
Tehat (6) tendenciajat ez a tovabbi tag nem tudja csokkenteni csak novelni.
Igy 2.5.4 tovabbra is igaz marad ha TS (D) != 0.

3. CONCLUSIONS

3.1 A fonti logika szerint az adodik hogy gyors, azaz 1.10 szerinti Futasi ido nem erheto el az N sakkfigura problema megoldasat vegzo A algoritmus implementaciojanak futtatasara.

3.2 Lathato (13)-bol hogy T0 = 0 eseten lehetne igaz a 2.1 tetszoleges D ertekre adott, allando k mellett. Ekkor viszont az A algoritmus implementacioja 0 ido alatt el tudna donteni egy adott lerakasrol hogy az helyes-e, de ez vegtelen gyors szamitogepet feltetelezne, amelyet jelenlegi ismereteink szerint lehetetlen megepiteni.

3.3 Nem hasznaltuk ki tovabba hogy vezert teszunk le a Sakktablara, igy 2.5.4 fennall akarmilyen hagyomanyos sakkfigura hasznalatara.