

## Zadanie č. 2

### I. Problém 3. – Úloha f)

Algoritmom slepého prehľadávania (do hĺbky) je možné nájsť (všetky) riešenia (v bežných výpočtových – čas a pamäť – podmienkach PC) iba pri šachovniciach do veľkosti 6x6, max 7x7. Implementujte tento algoritmus pre šachovnice s rozmermi 5x5 a 6x6 a skúste nájsť prvých 5 riešení pre každú šachovnicu tak, že pre šachovnicu 5x5 aj 6x6 si vyberte náhodne 5 východziech bodov (spolu teda 10 východziech bodov) s tým, že jeden z týchto bodov je (pre každú šachovnicu) ľavý dolný roh a pre každý z týchto bodov nájdite (skúste nájsť) prvé riešenie. V prípade, že ho v stanovenom limite nenájdete, signalizujte neúspešné hľadanie. V diskusii potom analyzujte pozorované výsledky.

### II. Opis riešenia

Riešenie sme navrhli pre všeobecný prípad úlohy Eulerovho koňa. Implementuje ho trieda `KnightsTour`. Jej metóda `init()` vytvorí front `Stack<Node>` pre uzly z okraju stromu hľadania a inicializuje ho začiatočným uzlom.

```
1  /**
2   * Vytvorí jednu úlohu Eulerov kôň s počiatočným stavom [x,y] a rozmerom
3   * šachovnice - dimension
4   */
5  public KnightsTour(int dimension, int x, int y)
6  {
7      this.dimension = dimension;
8      this.initX = x;
9      this.initY = y;
10     init(); // vytvor front a pridaj začiatočný uzol
11     solution = search(); // algoritmus hľadania do hĺbky
```

Riešenie úlohy získame prehľadávaním stavového priestoru do hĺbky založenom na princípe, že nové uzly sa vkladajú vždy na začiatok frontu. – metóda `search()`.

```
1  private Node search() {
2      while (System.currentTimeMillis() < endTime) // časový limit
3      {
4          if(stack.empty()) return null; // kontrola frontu
5
6          openNode = stack.pop(); // vybratie uzla z okraja
7
8          if(openNode.isFinalState()) return openNode; // skúška cieľu
9
10         // aplikuj všetky operátory
11         operator(openNode, 2, 1);
12
13         ...
```

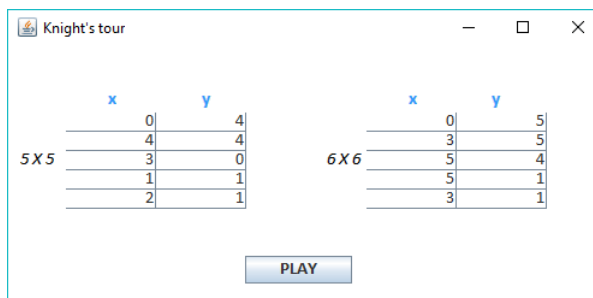
Pri hľadaní riešenia postupne vyberá uzly z frontu. Ak je prázdny riešenie neexistuje. Kontroluje či vybraný uzol neobsahuje cieľový stav.(t.j. všetky políčka boli prejdené). Nakoniec sa na uzol aplikuje 8 operátorov (posun koňa o dve polia jedným smerom horizontálne alebo vertikálne a jedno pole druhým smerom). Hľadanie má časový limit 15 sekúnd.

### III. Reprezentáciu údajov

Uzol je implementovaný triedou Node a obsahuje stav `int[dimension][dimension]state`. Údaj v matici reprezentuje poradie prejdienia daného políčka koňom. Nulové hodnoty značia, že kôň políčko ešte nenavštívil. Uzol ďalej obsahuje súradnice, kde sa práve kôň nachádza a počet jeho predošlých skokov.

### IV. Vstupné údaje

Vstupné údaje – súradnice východných bodov sa generujú náhodne, pričom jeden z týchto bodov je pre každú šachovnicu ľavý dolný roh.



### V. Analýza výstupu

Výstup programu je smerovaný do konzoly. Obsahuje počet prehľadaných stavov v priestore, čas behu hľadania (ak neprekročil limit) a cieľový stav = hľadané riešenie (ak sa našlo).

Príklady hľadání, pri ktorých sa nenašlo riešenie:

1# dimension: 5, start: [3, 4]  
number of states: 1829420  
time: 1222  
solution: not found

2# dimension: 6, start: [4, 0]  
number of states: 22252107  
time: exceeded  
solution: not found

V príklade #1 sa pri hľadaní front úplne vyprázdnil a riešenie sa nenašlo. Keďže je algoritmus hľadania do hĺbky pre náš konečný stavový priestor úplný – riešenie neexistuje. Z opakovaného testovania sme zistili, že riešenie neexistuje pre polovicu začiatkových stavov šachovnice 5x5 – sú to stavy, pri ktorých je súčet súradníc (indexovaných od nuly) nepárne číslo.

V príklade #2 sa riešenie nenašlo pretože vypršal časový limit 15 sekúnd. Riešenie sa nepodarilo nájsť ani pri limite 100 sekúnd a 144'801'157 prehľadaných stavoch.

3# dimension: 5, start: [0, 1]  
number of states: 1829420  
time: 1204  
solution: not found

4# dimension: 5, start: [4, 1]  
number of states: 1829420  
time: 1095  
solution: not found

-----  
5# dimension: 5, start: [0, 3]  
number of states: 1829420  
time: 1351  
solution: not found

-----  
6# dimension: 5, start: [4, 3]  
number of states: 1829420  
time: 1078  
solution: not found

Príklady hľadání #3 až #6 majú východzie body súmerne umiestnené na šachovnici 5x5 a neexistuje pre nich riešenie. To znamená, že budú rozvíjať rovnaký počet stavov. Tento fakt sa nám potvrdil aj z testovania. Všetky rozvíjajú 1829420 stavov.

## **VI. Zhodnotenie**

Náš program pre šachovnice veľkosti 5x5 našiel vždy riešenie ešte pred uplynutím časového limitu (ak riešenie existuje). Pre šachovnice 6x6 našiel vo väčšine testovacích prípadov riešenie pred uplynutím časového limitu.

Zrýchlenie behu programu by sa dalo dosiahnuť využitím efektívnejších dátových štruktúr. Časovo zložité je najmä kopírovanie stavu, reprezentovaného maticou celých čísiel, pri vytváraní nasledujúceho stavu.

Algoritmus hľadania do hĺbky má lepšiu priestorovú zložitost' ako prehľadávanie do šírky. Pre všetkých 10 scenárov je potrebná pamäť v desiatkach megabajtov.