Marcel Valent
Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4
xvalentm@stuba.sk
26. október 2021

# ZADANIE 2 – Eulerov kôň

Prednášajúci: Ing. Lukáš Kohútka, PhD.
Cvičiaci: Ing. Ivan Kapustík
Cvičenie: Streda 13:00

# Obsah

Marcel Valent
ID: 111658

# 1. Opis riešenia

V zadaní bolo špecifikované, že máme použiť algoritmus slepého prehľadávania do hĺbky. Túto podmienku som dodržal a využívam tento algoritmus s backtrackingom. V mojom programe sa najskôr zavolá funkcia ktorá zabezpečí vstupy od používateľa ( začiatočné súradnice, veľkosť, časový limit testovania). Následne sa zavolá funkcia ktorá mi vytvorí samotnú šachovnicu a nastaví sa v nej začiatočné políčko na hodnotu 0 (na voľných políčkach je -1). Následne sa zavolá samotná funkcia ktorá rieši pohyb koňa po šachovnici s využitím vyššie špecifikovaného algoritmu. Náš problém sa rieši až pokiaľ sa neprekročí počet krokov 10 miliónov, alebo nevyprší časový limit zadaný používateľom. Keď sa v tomto čase nájde riešenie tak nám program do konzole vytlačí samotnú šachovnicu, kde čísla znázorňujú číslo pohybu koňa po šachovnici.

# 2. Funkcie a reprezentácia údajov

**checkBounds(x, y, size, chessboard) -** funkcia na skontrolovanie, ci policko je volne a je v rozmeroch sachovnice

**printChessboard(size, chessboard) -** funkcia sluziaca na print sachovnice po dokonceni cesty kona

**randomCoord(size) -** funkcia na generovanie alebo zadávanie súradníc

**solver(size, x, y, timestart, limit) -** funkcia na generovanie sachovnice + print udajov do konzole

**solverHelp(size, chessboard, x, y, next_x, next_y, position, timestart, limit) -** funkcia na prehladavanie s backtrackingom

```
#definovanie moznych pohybov kona
possible_x_moves = [2, 1, -1, -2, -2, -1, 1, 2]
possible_y_moves = [1, 2, 2, 1, -1, -2, -2, -1]
```

Možné pohyby koňa mám reprezentované ako zmenu X a Y súradnice s tým, že Y súradnice a X súradnice sú každá vo svojom liste a sú popárované , čo znamená že musíme brať vždy z rovnakého indexu aj z Y listu aj z X listu.

```
#vygenerovanie šachovnice
chessboard = [[-1 for a in range(size)] for a in range(size)]
```

Samotnú šachovnicu mám reprezentovanú ako maticu. Tento typ som si vybral z dôvodu, že vďaka nemu môžem ľahko pristupovať k jednotlivým políčkam, keďže každá má svoju X a Y súradnicu.

Marcel Valent
ID: 111658

# 3. Testovanie

### Scenár 1

Veľkosť 6, začiatočná súradnica 0,0.

```
Zadaj veľkosť: 6
Zadaj časový limit hľadania (s): 15
Chceš randomne vygenerovať začiatočnú súradnicu?(y/n): n
Zadaj počiatočnú súradnicu x(0- 5 ): 0
Zadaj počiatočnú súradnicu y(0- 5 ): 0
0 15 6 25 10 13
33 24 11 14 5 26
16 1 32 7 12 9
31 34 23 20 27 4
22 17 2 29 8 19
35 30 21 18 3 28
Počet krokov: 248168
```

### Scenár 2

Veľkosť 6, náhodne vygenerované súradnice.

```
Zadaj veľkosť: 6
Zadaj časový limit hľadania (s): 15
Chceš randomne vygenerovať začiatočnú súradnicu?(y/n): y
Začiatočná súradnica x je:  4
Začiatočná súradnica y je:  5
Vypršal čas na hľadanie, program trval:  15 sekúnd
7 -1 9 12 5 -1
-1 13 6 -1 10 25
-1 8 11 18 1 4
14 21 16 3 24 19
-1 -1 23 20 17 2
22 15 -1 -1 0 -1
Počet krokov: 3327405
```

### Scenár 3

Veľkosť 7, začiatočná súradnica 0,0.

```
Zadaj veľkosť: 7
Zadaj časový limit hľadania (s): 15
Chceš randomne vygenerovať začiatočnú súradnicu?(y/n): n
Zadaj počiatočnú súradnicu x(0- 6 ): 0
Zadaj počiatočnú súradnicu y(0- 6 ): 0
Vypršal čas na hľadanie, program trval:  15 sekúnd
0 23 -1 7 -1 -1 14
-1 32 21 24 15 6 -1
22 1 26 -1 8 13 -1
-1 20 31 -1 25 16 5
-1 27 2 29 18 9 12
-1 30 19 -1 11 4 17
-1 -1 28 3 -1 -1 10
Počet krokov: 2843886
```

```
Zadaj veľkosť: 7
Zadaj časový limit hľadania (s): 50
Chceš randomne vygenerovať začiatočnú súradnicu?(y/n): n
Zadaj počiatočnú súradnicu x(0- 6 ): 0
Zadaj počiatočnú súradnicu y(0- 6 ): 0
0 37 30 7 18 35 14
31 28 19 36 15 6 17
38 1 32 29 8 13 34
27 24 39 20 33 16 5
40 21 2 25 44 9 12
23 26 47 42 11 4 45
48 41 22 3 46 43 10
Počet krokov: 7151178
```

**Scenár 4**

Veľkosť 7, náhodne vygenerované súradnice.

```
Zadaj veľkosť: 7
Zadaj časový limit hľadania (s): 15
Chceš randomne vygenerovať začiatočnú súradnicu?(y/n): y
Začiatočná súradnica x je:  4
Začiatočná súradnica y je:  6
Vypršal čas na hľadanie, program trval:  15 sekúnd
-1 -1 34 7 24 -1 -1
31 -1 23 18 33 6 -1
22 19 32 25 8 1 -1
-1 30 17 20 3 12 5
-1 21 26 29 14 9 2
27 -1 -1 16 11 4 13
-1 -1 28 -1 0 15 10
Počet krokov: 2759867
```

**Scenár 5**

Veľkosť 7, začiatočná súradnica 6,6.

```
Zadaj veľkosť: 7
Zadaj časový limit hľadania (s): 15
Chceš randomne vygenerovať začiatočnú súradnicu?(y/n): n
Zadaj počiatočnú súradnicu x(0- 6 ): 6
Zadaj počiatočnú súradnicu y(0- 6 ): 6
44 35 24 9 12 3 48
25 22 45 4 47 8 11
34 43 36 23 10 13 2
37 26 21 46 5 16 7
42 33 40 29 18 1 14
27 38 31 20 15 6 17
32 41 28 39 30 19 0
Počet krokov: 133015
```

### 3.1. Zhodnotenie testovania

Ako sme si mohli všimnúť tak časová aj kroková náročnosť pri 7x7 šachovnici je omnoho vyššia ako pri 6x6 šachovnici. Algoritmus však nie je schopný nájsť cestu z niektorých súradníc do časového limitu, ale po zvýšení časového limitu už cestu našlo (scenár 3). Prekvapením bol scenár číslo 5, kde mi cestu z týchto súradníc našlo takmer instantne a na relatívne malý počet krokov.

## 4. Možnosti optimalizácie

Zadaný algoritmus sa už nedá ďalej optimalizovať, jediný potenciál na optimalizáciu vidím v použití inej postupnosti krokov koňa. Táto optimalizácia však môže síce niektoré riešenia zrýchliť, iné však môže naopak spomaliť. Celkové zrýchlenie programu by sme mohli dosiahnuť iba pomocou využitia iného algoritmu hľadania, alebo za použitia heuristík. Tieto som ale v mojom zadaní nemal, tak som sa s nimi príliš nezaoberal.

# 5. Implementačné prostredie

Ako implementačné prostredie využívam Python a ako IDE používam Pycharm od spoločnosti JetBrains. Implementačné prostredie určite vplýva na rýchlosť programu, keďže optimalizovaný program v jazyku C by mi určite zbehol rýchlejšie ako v jazyku Python, avšak jazyk Python ponúka jednoduchšiu prácu s pamäťou. Z tohto dôvodu som sa rozhodol pre tento jazyk. Čo sa týka používania môjho programu, tak používanie je veľmi jednoduché.

# Záver

Cieľom môjho zadania bolo využitie **slepého prehľadávania do hĺbky** pri hľadaní problému Eulerovho koňa na šachovniciach 6x6 a 7x7. Myslím, že úlohu zadania som splnil, keďže môj program vie nájsť riešenia pre už vyššie spomenuté veľkosti šachovnice. Na väčších šachovniciach už bude časová efektivita tragická z dôvodu použitia slepého hľadania, preto som ju ani neimplementoval a netestoval (po odstránení niektorých podmienok sa dá pustiť aj NxN veľkosť šachovnice).