Meno: Marko Stahovec
ID: 110897

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

ZADANIE 2 – Eulerov Kôň (Warndorffove pravidlo)

Meno a priezvisko: Marko Stahovec

Dátum vypracovania: 17.10.2021

Cvičiaci: Ing. Boris Slíž

Meno: Marko Stahovec
ID: 110897

Meno: Marko Stahovec
ID: 110897

1. Zadanie

Našou úlohou bolo prejsť šachovnicu legálnymi ťahmi šachového koňa tak, aby každé políčko šachovnice bolo prejdené (navštívené) práve raz. Riešenie sme mali navrhnúť tak, aby bolo možné problém riešiť pre štvorcové šachovnice rôznych veľkostí (minimálne od veľkosti 5 x 5 do 20 x 20) a aby cestu po šachovnici bolo možné začať na ľubovoľnom políčku.

Moje zadanie bolo rozšírené o heuristiku, ktorá sa nazýva Warndorffove pravidlo. Túto heuristiku som implementoval do slepého prehľadávania do hĺbky a bližšie ju popíšem v bode 2.

2. Opis riešenia

Moje riešenie funguje na princípe **prehľadávania do hĺbky**, keďže funkcia jump() vykonáva rekurzívne kroky do hĺbky a pokúša sa nájsť riešenie priamo.
Prvou funkciou, v ktorej môj algoritmus začína, vyzerá takto:

```
def place_knight(size, x, y, run):  # menu for an algorithm
    global turn
    turn = 0  # reset for number of turns
    board = [[None] * size for i in range(size)]  # fill the board with None
    moves = [[2, 1], [1, 2], [-1, 2], [-2, 1], [-2, -1], [-1, -2], [1, -2], [2, -1]]  # list of all possible moves
    jump(1, x, y, board, moves, run)  # start
```

Argumenty tejto funkcie sú nasledovné:

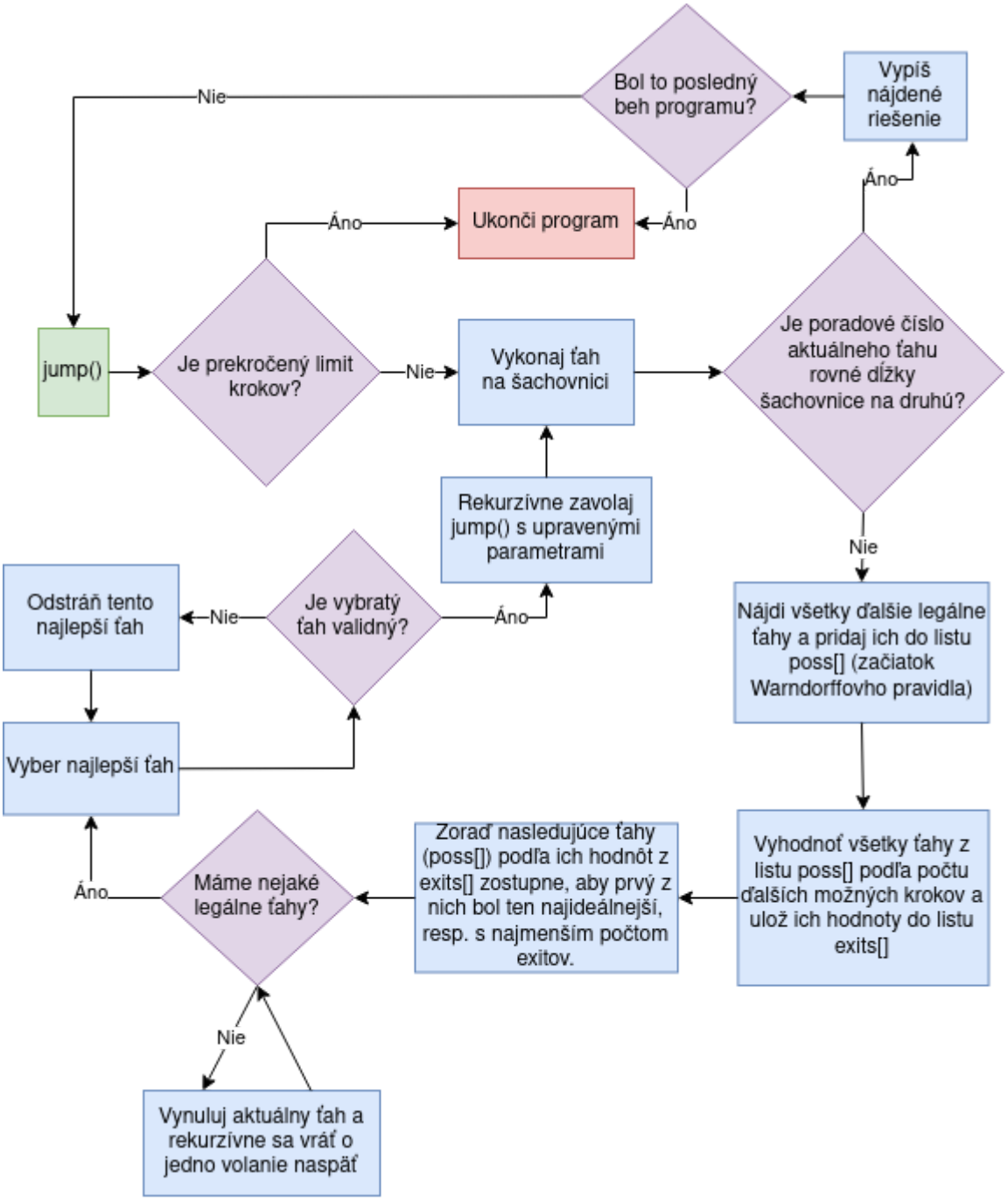
- **size** - veľkosť šachovnice
- **x** - počiatočná x-ová súradnica
- **y** - počiatočná y-ová súradnica
- **run** - poradové číslo aktuálneho behu

Na začiatku pristupujeme ku globálnej premennej s názvom **turn**, v ktorej sa počas behu udržiava **počet krokov**, ktoré algoritmus vykonal. Ak nastane situácia, že program prekročí istú hranicu (napr, 2 000 000), tak je ukončený príkazom exit(5).

Meno: Marko Stahovec
ID: 110897

V premennej **board** sa inicializuje šachovnica na rozmery **size*size** a vyplní sa hodnotou None. V liste **moves[]** je zoznam všetkých možných pohybov, ktoré môže kôň logicky vykonať. Následne je už len volanie pre našu algoritmickú funkciu s názvom **jump** a parametrami, ktoré sú vysvetlené vyššie.

Stručný návrh **fungovania algoritmu** (funkcie jump()) je naznačený vo flow-charte nižšie:



Vypisovanie riešenia je zabezpečené vnoreným for cyklom vo for cykle, a teda výpisom 2D listu v premennej **board**, v ktorej sa nachádzajú všetky kroky. Hodnoty None v premennej board sa nahrádzajú **číslom**, ktoré korešponduje s **poradovým číslom ťahu**, ktoré kôň vykonal.

Počet behov programu je daný konštantou **ROUNDS**, ktorej hodnotu môžeme kedykoľvek zmeniť v kóde.

Usporadúvanie ťahov je vykonané vložením do slovníka a zoradením podľa počtu exitov pre každý nasledujúci ťah, čo zaručí, že prvý ťah, ktorý si vytiahne cyklus z poľa sorted_moves bude ťah s najmenším počtom exitov, a teda heuristika Warndorffovho pravidla je dodržaná.

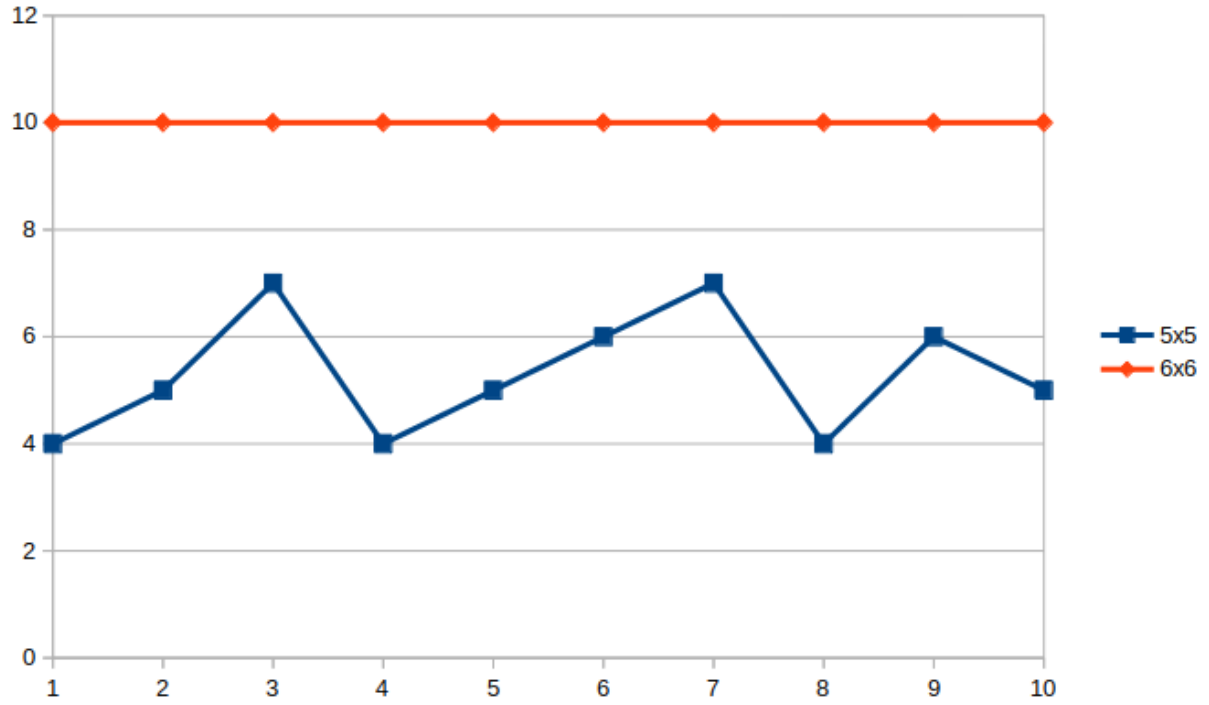
3. Testovanie

V tejto časti zhrniem poznatky z testovania, ktoré som vykonal na svojom programe. Moje riešenie som testoval na **rôznych veľkostiach šachovnice** s počtom šachovníc 10. Následne som vyhodnocoval, koľko riešení z daných 10 sa môjmu programu podarilo nájsť. Každú veľkosť šachovnice som otestoval **10-krát**, začínajúc od rozmeru 5x5. Neskôr v testovaní sú spomenuté porovnania väčších šachovníc až do rozmerov 30x30, i keď mnou navrhnuté riešenie je schopné riešiť problémy na šachovnici o rozmeroch vyše 70x70.
Na záver som otestoval aj **rýchlosť pri hľadaní jedného riešenia** na rovnakých rozmeroch.

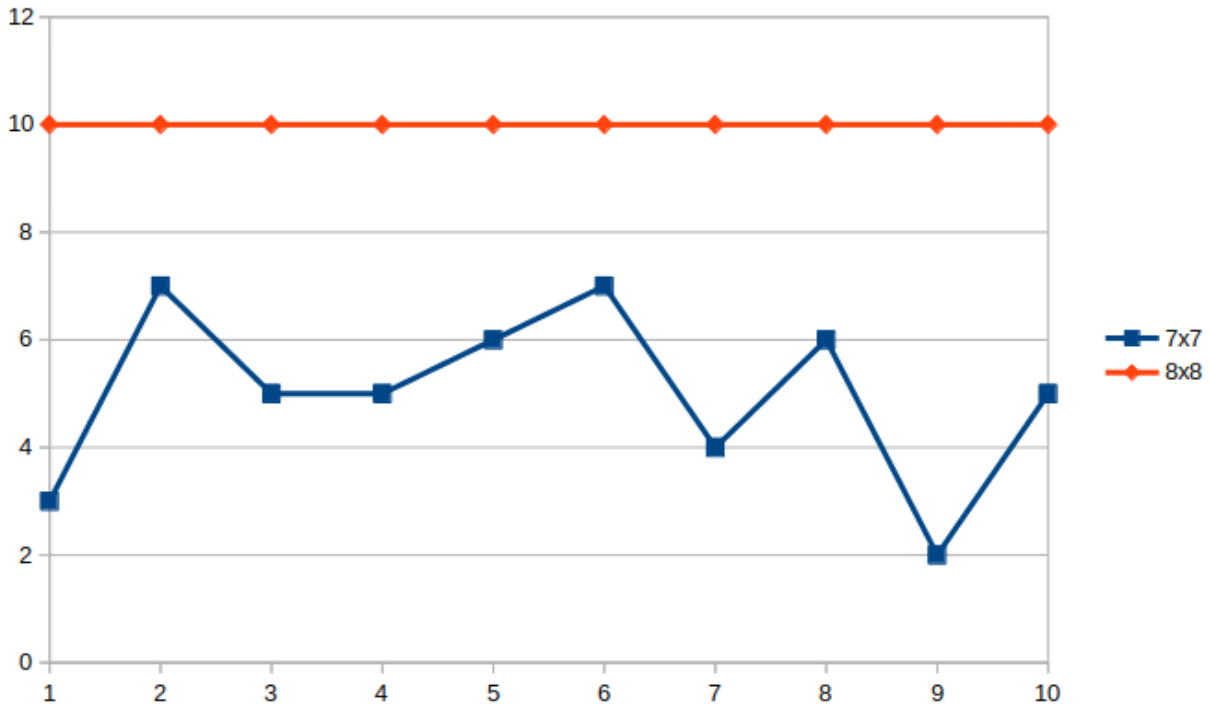
Počiatočné súradnice boli **generované náhodne** s podmienkou, aby boli všetky rôzne.
Premenná ROUNDS bola pri menších šachovniciach nastavená na 1 000 000, pri porovnaní 7x7 a 8x8 na 2 000 000, 12x12 a 15x15 na 3 000 000 a 20x20 a 30x30 na 4 000 000.

Graf pre testovanie 5x5 a 6x6:

Meno: Marko Stahovec
ID: 110897

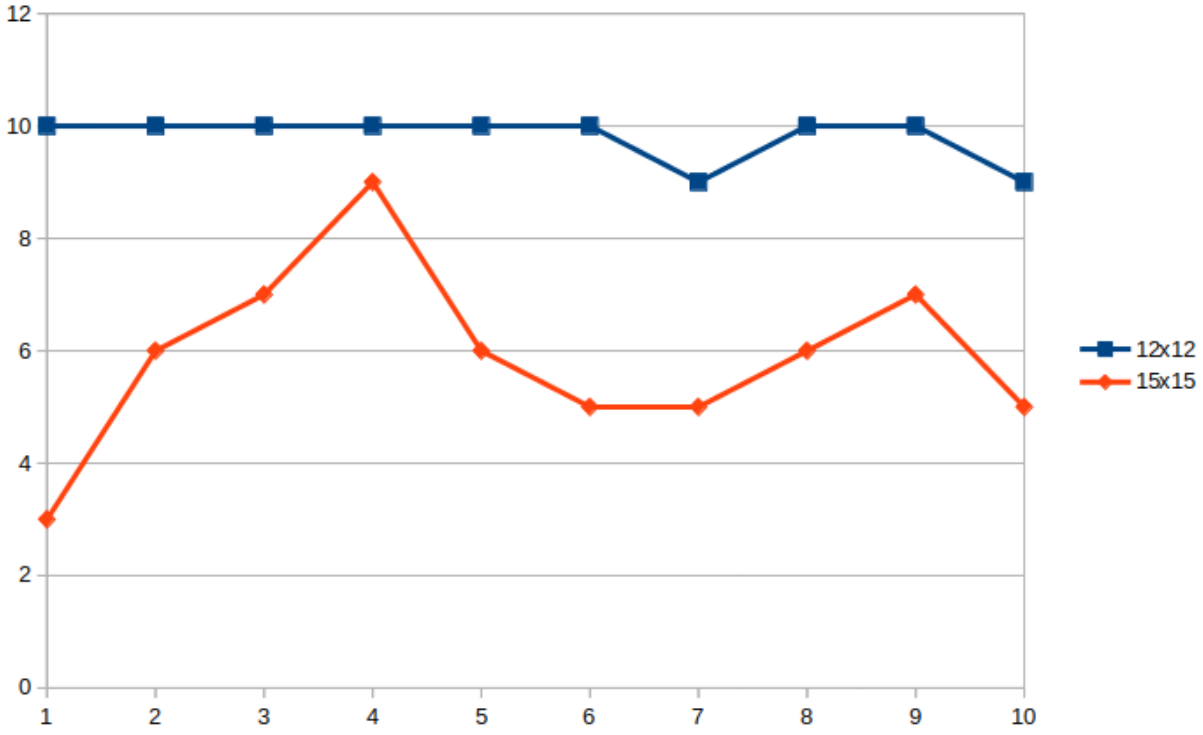


Na x-ovej osi je číslo behu a na y-ovej osi je počet nájdených riešení z 10 rôznych poriatočných súradníc. Ako môžeme vidieť na grafe, tak menšie šachovnice **nevyhovujú** tomuto algoritmu, keďže ho prílišne obmedzujú na počet možností. Môžeme predpokladať, že problém **nie je riešiteľný** pre konkrétne dvojice súradníc, keďže sa podľa všetkého generovali opakovane v každom behu v rôznom počte.



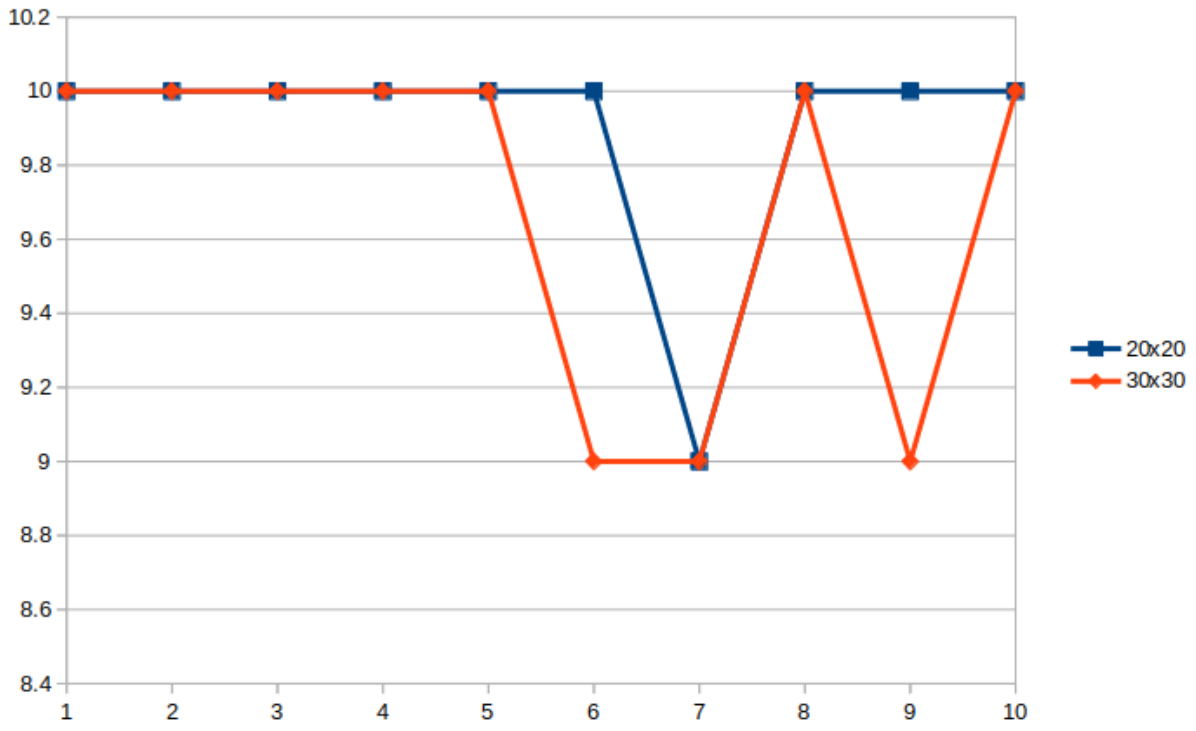
Meno: Marko Stahovec
ID: 110897

Ako môžeme vidieť na grafe porovnania 7x7 a 8x8, tak problémy na šachovnici o rozmeroch 7x7 mali podobný **problém** ako v predchádzajúcom teste šachovnice 5x5. Z toho usudzujem, že moja **predchádzajúca hypotéza** o náraste počtu riešení pri vyšších šachovniciach môže platiť, no nárast nie je lineárny a kolíše práve v takýchto špecifických prípadoch, kde sa mení až na pokles. Tento fenomén nie je až takým prekvapením, keďže rovnaký princíp je pozorovateľný aj v podobnom šachovnicovom probléme, ktorý sa nazýva Queen's problem.



Fenomén spomenutý v predchádzajúcom meraní sa nám opäť potvrdil, no tentoraz už môžeme s istotou preformulovať našu hypotézu. Tá znie, že kôň **má problém ukončiť cestu na šachovniciach s nepárnym rozmerom**. Toto tvrdenie mi bolo potvrdené aj prácou, ktorej zdroj je uvedený nižšie [1].

Meno: Marko Stahovec
ID: 110897



Dĺžka vykonávania bola odhadom kratšia ako pol sekundy aj pri rozmeroch 20x20 a 30x30, no v prípadoch, kedy bol potrebný intenzívny backtracking, bol čas vykonávania badateľný dlhší.

Každopádne, na záver testovania som meral aj **časovú výkonnosť** presnými hodnotami. Premenná **turn** bola nastavená na **4 000 000** pre všetky rozmery, aby boli zaručené férové podmienky.

	5x5	6x6	7x7	8x8	12x12	15x15	20x20	30x30
1	0.0019	0.0026	39.2141	0.0042	0.0113	0.0076	0.0141	0.0421
2	0.0028	0.0026	0.0035	0.0022	0.0035	40.7141	0.0142	0.0199
3	0.0018	0.0012	0.0026	0.005	0.011	0.0178	0.0339	0.0526
4	0.0006	0.0043	39.0152	0.0015	0.058	0.0056	0.0118	0.0577
5	17.6654	0.0009	2.8963	0.0048	0.0101	0.0055	0.0352	0.0236
6	0.0009	0.0026	2.3972	0.0047	0.0059	0.008	0.0346	0.0297
7	0.001	0.0008	0.0032	0.0029	0.0044	0.0178	0.0329	0.0518
8	17.4316	0.0026	39.0442	0.0021	0.0111	40.3325	0.0355	0.0284
9	9.7552	0.0008	0.0025	0.0048	0.0035	0.0157	0.0128	0.031
10	0.0044	0.0012	0.0033	0.0026	0.0099	0.0132	0.036	0.0514

Žltou farbou sú vyznačené behy, v ktorých nastalo dosiahnutie hraničnej hodnoty premennej turn, čo znamenalo **ukončenie** programu. **Modrou** sú zvýraznené **najrýchlejšie** behy z meraných 10 pokusov. Z tejto tabuľky je zrejmé, že nepárne

veľkosti šachovnice nechutia nášmu algoritmu. Takisto si môžeme všimnúť zaujímavosť, že zacyklenie programu a následné ukončenie programu dlhšie pri väčších rozmeroch šachovnice.
Taktiež pri behoch č.5 a 6 pre rozmer 7x7 pozorujeme výrazne zvýšený čas behu programu, keďže bol v daných prípadoch potrebný intenzívny backtracking a prehľadávanie iných možností.

4. Zhodnotenie testovania

Z testovania vyplýva niekoľko dôležitých poznatkov, no v prvom rade je potrebné poznamenať, že **daná heuristika naozaj pomáha a urýchľuje hľadanie riešení** pre tento problém. Úspešnosť hľadania riešení je v mojom programe znásobená prítomnosťou **backtrackingu**, ktorý je možno vidieť napr. pri šachovnici o rozmeroch 7x7 a počiatočnými súradnicami [2,2]. Tento prípad odprezentujem aj na cvičení.

Taktiež sme zistili, že Eulerov kôň má **dramaticky nižší počet riešení** v prípadoch, ak je rozmer šachovnice **nepárny** a že veľkosť šachovnice len minimálne obmedzuje účinnosť algoritmu.

Časová výkonnosť prirodzene **klesá pri väčších rozmeroch šachovnice**, no pri rozmere 30x30 sa rýchlosť algoritmu pohybuje **pod jednu desatinu sekundy**, čo je stále mimoriadne rýchle.

5. Záver

Myslím si, že moje riešenie je viac ako vyhovujúce, keďže je **nenáročné na pochopenie a mimoriadne efektívne** aj napriek faktu, že je naprogramované v jazyku Python. Riešenia hľadá **spoľahlivo** a je obohatené aj o funkcie, ktoré nebolo potrebné implementovať.

6. Zdroje

[1]:
https://books.google.sk/books?id=eXWUDwAAQBAJ&pg=PA173&lpg=PA173&dq=warnsdorff%27s+rule+for+odd+sizes&source=bl&ots=1T33T-Ctcq&sig=ACfU3U0fkJnayHNSY5UuvsDqioqyL9T0dQ&hl=sk&sa=X&ved=2ahUKEwj6oIDvhtzzAhWWgf0HHUvuA-MQ6AF6BAgrEAM#v=onepage&q=warnsdorff's%20rule%20for%20odd%20sizes&f=false