

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Umelá Inteligencia

Zadanie 2 – Eulerov kôň

ZS 2021/2022

Samuel Kováč

Študijný program: B-INFO4
Vyučujúci: Ing. Ivan Kapustík
Cvičenia: Streda 13:00
Október 2021

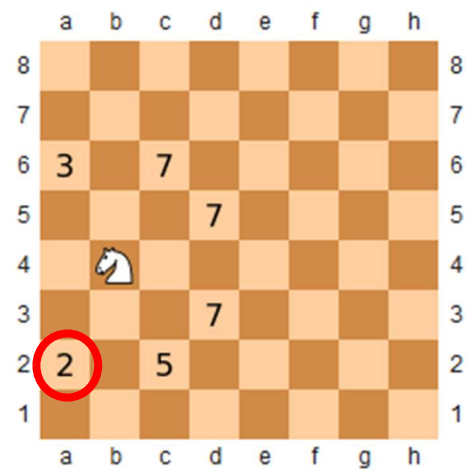
Zadanie :

Úlohou je prejsť šachovnicu legálnymi ťahmi šachového koňa tak, aby každé políčko šachovnice bolo prejdené (navštívené) práve raz. Riešenie treba 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 východnom políčku.

Riešil som zadanie **g**).

Riešenie :

Na riešenie daného problému som využil heuristiku s názvom *Warnsdorff's rule* z roku 1823. Heuristika spočíva v tom, že koň vždy skočí na políčko s najmenším počtom ďalších možných skokov.



V danej pozícii podľa použitej heuristiky skočí jazdec na políčko **a2**

Implementácia heuristiky :

Najprv zistíme počty ďalších možných skokov zo všetkých dostupných smerov. V prípade že nájdeme viac políčok z rovnakým najmenším počtom skokov, využijeme rekurzívne prehľadávanie do hĺbky. To znamená že v prípade voľby nesprávneho skoku sa jazdec vráti späť na posledné rozhodovanie sa a vyberie ďalšiu možnosť. V programe je podľa zadania implementovaný časový limit pre nájdenie riešenia 15 sekúnd, no pri testovaní som zistil že vďaka efektívnosti tejto heuristiky je program limitovaný skôr pamäťou ako časom. Najväčší možný rozmer šachovnice ktorú program úspešne vyrieši je 31x31.

Reprezentácia údajov :

Šachovnicu v mojom programe reprezentujem triedou *Sachovnica*. Vstupné parametre šachovnice sú :

1. dĺžka strany n
2. štartovacia pozícia jazdca - súradnica X
3. štartovacia pozícia jazdca - súradnica Y

Šachovnica si v sebe uchováva aj momentálne číslo kroku jazdca, a tiež čas v ktorom sa spustilo hľadanie riešenia.

Pri inicializácii tejto triedy program vytvorí samotnú šachovnicu ako dvojrozmerný list, skontroluje štartovacie súradnice a nastaví jazdca na správnu pozíciu. Na konci inicializácie sa aj automaticky spustí hľadanie riešenia danej šachovnice.

Jazdec má vždy 8 možností skoku a teda 8 operátorov. Ukladám ich ako dvojrozmerný list :

```
posunutia = [[1,2],[1,-2],[2,1],[2,-1],[-1,2],[-1,-2],[-2,1],[-2,-1]]
```

Testovanie

Najmenšia možná vyriešiteľná šachovnica má rozmer 5x5 :

Nájdené riešenie :				
1	12	25	18	3
22	17	2	13	24
11	8	23	4	19
16	21	6	9	14
7	10	15	20	5

Príklad riešenia 8x8 šachovnice :

Nájdené riešenie :							
1	4	57	20	47	6	49	22
34	19	2	5	58	21	46	7
3	56	35	60	37	48	23	50
18	33	38	55	52	59	8	45
39	14	53	36	61	44	51	24
32	17	40	43	54	27	62	9
13	42	15	30	11	64	25	28
16	31	12	41	26	29	10	63

Hlavnou výhodou mojej implementácie je časová náročnosť. Keďže veľké šachovnice majú obrovský počet možných riešení, použitá heuristika ich dokáže ľahko nájsť aj bez zdĺhavého prehľadávania do hĺbky. Nevýhodou je ale rekurzívne hľadanie ktoré v prípade programovacieho jazyku Python zvládne najväčší rozmer šachovnice len 31 x31.