

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Tibor Vanek

## Zadanie 2 – **Eulerov kôň**

**g)**

Predmet: Umelá Inteligencia

Čas cvičenia: Štvrtok 14:00

Cvičiaci: Ing. Ivan Kapustík

## Zadanie

### EULEROV KÔŇ

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 východizom políčku.

Jedno z mnohých riešení Eulerovho koňa s šachovnicou o rozmeroch 5x5 je napríklad toto:

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

V tomto príklade sa úspešná cesta Eulerovho koňa začala v ľavom hornom rohu (označenom číslom 1) a skončila uprostred šachovnice (na políčku označenom číslom 25), čísla označujú poradové číslo ťahu.

### STAV, OPERÁTORY, ...

Aktuálny stav najlepšie reprezentuje šachovnica. Na začiatku sú všetky jej políčka vynulované a postupne sa zaplňa číslami, ktoré zodpovedajú príslušnému skoku koňa. Na reprezentáciu uzla je potrebné pridať pre každý skok aj poradové číslo operátora alebo zoznam možných, ešte nevyužitých skokov, aby program pri návrate vedel, akou alternatívou pokračovať, prípadne sa vrátiť ešte o ďalší skok naspäť.

Kôň má vo všeobecnosti 8 možností skoku, ak nie je limitovaný okrajom šachovnice alebo už použitým miestom. To znamená, že existuje 8 operátorov, ktoré je možné označiť napríklad

$(1, 2), (1, -2), (2, 1), (2, -1),$

$(-1, 2), (-1, -2), (-2, 1)$  a  $(-2, -1),$

kde prvé číslo znamená posun od aktuálnej pozície v riadku a druhé v stĺpci.

Algoritmus je jednoducho klasický algoritmus prehľadávania do hĺbky, kde sú všetky štruktúry na ukladanie uzlov nahradené (rozšírenou) šachovnicou.

## POZNÁMKA 1

Pre problém Eulerovho kôňa treba v oboch úlohách uvažovať s tým, že pre niektoré východzie políčka a niektoré veľkosti šachovnice riešenie neexistuje. Program preto treba navrhnuť a implementovať tak, aby sa v prípade, že do určitého času, resp. počtu krokov riešenie nenájde, zastavil a signalizoval neúspešné hľadanie. Maximálny počet krokov, resp. maximálny čas hľadania by preto mal byť ako jeden zo vstupných (voliteľných) parametrov programu. Pre toto zadanie a testovacie príklady je odporúčaný maximálny počet krokov jeden až desať miliónov, resp. maximálny čas 15 sekúnd.

## Opis riešenia

Moje riešenie tohto zadania spočíva v použití Warnsdorffovho pravidla a rekurzívneho prehľadávania do hĺbky. Je vcelku jednoduché.

Jednoduchý **popis algoritmu**:

1. Špecifikujem si veľkosť šachovnice a štartovú pozíciu.
2. Vytvorím si šachovnicu veľkosti  $M \times M$  cez funkciu `createBoard()` a označím v tomto poli začiatočnú pozíciu.
3. Nájdem možné kroky z aktuálnej pozície, ktoré môže urobiť kôň, nie sú mimo šachovnice a ešte som ich nenavštívil. Ak som žiadne nenašiel, vrátim sa v rekurzii o krok späť a pokračujem.
4. Pozriem sa, koľko krokov sa dá vykonať z každého a vyberiem minimum.
5. Na vybrané políčko skočím. Opakujem krok 3 až dokiaľ sa hĺbka vnorenia nerovná druhej mocnine dĺžky/šírky šachovnice (`depth == size*size`).

Ako je uvedené v poznámke 1, pre niektoré začiatočné body neexistuje riešenie a preto som implementoval aj pravidlo, že ak sa tento algoritmus nevykoná do 15 sekúnd, hľadanie skončí.

Na toto zadanie som použil prostredie PyCharm, programovací jazyk **Python** (3.10) a používam knižnicu *time* na meranie času vykonania programu. Šachovnicu ukladám ako dvoj-rozmerné pole (list listov). Jednotlivé pozície v šachovnici ukladám ako „Tuple“ v tvare (x, y) – napr. (0, 0).

## Zhodnotenie a testovanie

Testoval som viacero veľkostí šachovnice a rôzne začiatočné body, pre niektoré algoritmus nenašiel riešenie do časového limitu. Najväčšia možná šachovnica, na ktorej algoritmus funguje je 31x31 kvôli limitu rekurzií v Pythone.

Efektivita algoritmu by mohla byť zmenená napr. zmenením poradia prvkov v tomto liste (kroky sa totiž prechádzajú v tomto poradí):

```
legal_steps = [  
    (1, 2), (1, -2), (2, 1), (2, -1),  
    (-1, 2), (-1, -2), (-2, 1), (-2, -1),  
]
```

alebo náhodným výberom kroku;

Ak sa minimum z počtu možných krokov zhoduje v 2 alebo viacerých možnostiach, prípadne iný zlepšovací algoritmus na efektívnejší výber.

**POZOR: súradnice štartového bodu v testovaní uvádzam od 0 po MAX-1, rovnako ako sú reprezentované v programe.**

Príklady testov:

**5x5, štart (0,0)**

[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]

Uplynutý čas: 0.0009992122650146484 s

**5x5, štart (1,1)**

[23, 6, 17, 12, 25]

[16, 1, 24, 5, 10]

[7, 22, 11, 18, 13]

[2, 15, 20, 9, 4]

[21, 8, 3, 14, 19]

Uplynutý čas: 0.00099945068359375 s

**5x5, štart (2,4)**

[23, 10, 5, 14, 25]

[4, 15, 24, 11, 6]

[9, 22, 13, 18, 1]

[16, 3, 20, 7, 12]

[21, 8, 17, 2, 19]

Uplynutý čas: 0.11746740341186523 s

**5x5, štart (1,2)**

Vypršal časový limit 15s

**6x6, štart (0,0)**

[1, 32, 9, 18, 3, 36]  
[10, 19, 2, 33, 28, 17]  
[31, 8, 29, 16, 35, 4]  
[20, 11, 34, 27, 24, 15]  
[7, 30, 13, 22, 5, 26]  
[12, 21, 6, 25, 14, 23]

Uplynutý čas: 0.0009996891021728516 s

**7x7, štart (0,0)**

[1, 4, 15, 28, 35, 6, 49]  
[14, 29, 2, 5, 46, 27, 34]  
[3, 16, 45, 36, 33, 48, 7]  
[30, 13, 32, 47, 22, 37, 26]  
[17, 44, 23, 38, 41, 8, 21]  
[12, 31, 42, 19, 10, 25, 40]  
[43, 18, 11, 24, 39, 20, 9]

Uplynutý čas: 0.00099945068359375 s

**8x8, štart (0,0)**

[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]

Uplynutý čas: 0.0009996891021728516 s

**8x8, štart (1,0)**

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

Uplynutý čas: 0.0010030269622802734 s

**8x8, štart (1,1)**

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

Uplynutý čas: 0.0019996166229248047 s

**8x8, štart (1,2)**

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

Uplynutý čas: 0.001999378204345703 s

**8x8, štart (2,2)**

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

Uplynutý čas: 0.0019986629486083984 s

**8x8, štart (7,7)**

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

Uplynutý čas: 0.001998424530029297 s

**8x8, štart (6,6)**

[29, 10, 39, 34, 27, 12, 41, 44]

[38, 35, 28, 11, 40, 43, 26, 13]

[9, 30, 37, 58, 33, 64, 45, 42]

[36, 55, 32, 63, 60, 47, 14, 25]

[31, 8, 59, 48, 57, 24, 61, 46]

[54, 5, 56, 23, 62, 49, 18, 15]

[7, 22, 3, 52, 17, 20, 1, 50]

[4, 53, 6, 21, 2, 51, 16, 19]

Uplynutý čas: 0.002001047134399414 s

**8x8 (štart 0,7)**

[22, 63, 6, 47, 20, 49, 4, 1]

[7, 46, 21, 64, 5, 2, 19, 34]

[62, 23, 56, 37, 48, 35, 50, 3]

[45, 8, 61, 52, 55, 38, 33, 18]

[24, 53, 44, 57, 36, 51, 14, 39]

[9, 60, 27, 54, 43, 40, 17, 32]

[28, 25, 58, 11, 30, 15, 42, 13]

[59, 10, 29, 26, 41, 12, 31, 16]

Uplynutý čas: 0.0019989013671875 s

**8x8, štart (2,5)**

[58, 5, 28, 25, 64, 3, 30, 15]

[27, 24, 59, 4, 29, 14, 61, 2]

[6, 57, 26, 63, 60, 1, 16, 31]

[23, 38, 55, 52, 41, 62, 13, 50]

[56, 7, 42, 39, 54, 51, 32, 17]

[37, 22, 53, 46, 43, 40, 49, 12]

[8, 45, 20, 35, 10, 47, 18, 33]

[21, 36, 9, 44, 19, 34, 11, 48]

Uplynutý čas: 0.003997802734375 s

**8x8, štart (6,3)**

[44, 11, 64, 59, 46, 13, 38, 53]

[63, 60, 45, 12, 51, 54, 47, 14]

[10, 43, 58, 61, 48, 39, 52, 37]

[57, 62, 27, 42, 55, 50, 15, 40]

[24, 9, 56, 49, 26, 41, 36, 31]

[3, 6, 25, 28, 35, 32, 19, 16]

[8, 23, 4, 1, 18, 21, 30, 33]

[5, 2, 7, 22, 29, 34, 17, 20]

Uplynutý čas: 0.0019986629486083984 s

**9x9, štart (0,0)**

[1, 4, 61, 18, 69, 6, 37, 20, 35]  
[62, 17, 2, 5, 64, 19, 34, 7, 38]  
[3, 60, 63, 68, 33, 70, 39, 36, 21]  
[16, 67, 32, 75, 48, 65, 22, 77, 8]  
[31, 56, 59, 66, 81, 76, 71, 40, 23]  
[58, 15, 80, 49, 74, 47, 78, 9, 72]  
[55, 30, 57, 46, 79, 50, 73, 24, 41]  
[14, 45, 28, 53, 12, 43, 26, 51, 10]  
[29, 54, 13, 44, 27, 52, 11, 42, 25]  
Uplynutý čas: 0.001998424530029297 s

**10x10, štart (0,0)**

[1, 4, 97, 32, 91, 6, 73, 30, 71, 8]  
[46, 33, 2, 5, 98, 31, 92, 7, 74, 29]  
[3, 100, 45, 96, 83, 90, 75, 72, 9, 70]  
[34, 47, 84, 99, 76, 93, 68, 89, 28, 63]  
[85, 18, 95, 44, 87, 82, 77, 62, 69, 10]  
[48, 35, 86, 81, 94, 59, 88, 67, 64, 27]  
[17, 80, 19, 58, 43, 78, 61, 56, 11, 66]  
[20, 49, 36, 79, 60, 57, 42, 65, 26, 55]  
[37, 16, 51, 22, 39, 14, 53, 24, 41, 12]  
[50, 21, 38, 15, 52, 23, 40, 13, 54, 25]  
Uplynutý čas: 0.0019991397857666016 s

**20x20, štart (0,0)**

[1, 4, 199, 400, 207, 6, 205, 376, 209, 8, 291, 294, 211, 10, 289, 214, 143, 12, 145, 216]  
[198, 399, 2, 5, 204, 377, 208, 7, 292, 375, 210, 9, 290, 295, 212, 11, 222, 215, 142, 13]  
[3, 200, 393, 386, 397, 206, 369, 378, 373, 358, 293, 364, 299, 288, 221, 296, 213, 144, 217, 146]  
[394, 197, 398, 203, 392, 385, 388, 357, 370, 365, 374, 355, 340, 331, 298, 287, 220, 223, 14, 141]  
[201, 38, 391, 396, 387, 368, 383, 372, 379, 356, 359, 332, 363, 300, 339, 224, 297, 226, 147, 218]  
[196, 395, 202, 347, 384, 389, 380, 351, 366, 371, 362, 341, 354, 333, 330, 231, 286, 219, 140, 15]  
[37, 252, 39, 390, 267, 348, 367, 382, 319, 350, 353, 360, 327, 338, 301, 334, 225, 232, 227, 148]  
[40, 195, 268, 253, 346, 381, 318, 349, 352, 361, 326, 337, 342, 329, 324, 285, 230, 277, 16, 139]  
[187, 36, 251, 192, 269, 266, 345, 320, 313, 308, 343, 328, 325, 302, 335, 278, 233, 242, 149, 228]  
[194, 41, 188, 265, 254, 317, 270, 263, 344, 321, 314, 309, 336, 323, 284, 241, 276, 229, 138, 17]  
[35, 186, 193, 250, 191, 264, 255, 316, 307, 312, 281, 322, 303, 310, 279, 234, 243, 150, 239, 134]  
[42, 189, 92, 185, 256, 249, 172, 271, 262, 315, 306, 311, 280, 283, 304, 275, 240, 135, 18, 137]  
[91, 34, 181, 190, 173, 184, 257, 248, 171, 272, 261, 282, 305, 274, 235, 244, 151, 238, 133, 122]  
[180, 43, 174, 93, 182, 177, 166, 111, 258, 247, 160, 273, 260, 245, 154, 237, 128, 121, 136, 19]  
[33, 90, 179, 176, 167, 94, 183, 170, 165, 110, 259, 246, 159, 236, 127, 152, 155, 132, 123, 120]  
[44, 175, 78, 89, 178, 169, 164, 95, 112, 161, 158, 107, 126, 153, 156, 129, 124, 117, 20, 103]  
[79, 32, 69, 168, 81, 88, 73, 162, 83, 96, 109, 114, 157, 106, 125, 116, 131, 104, 119, 56]  
[68, 45, 80, 77, 72, 163, 82, 87, 74, 113, 84, 97, 108, 115, 130, 105, 118, 57, 102, 21]  
[31, 70, 47, 66, 29, 76, 49, 64, 27, 86, 51, 62, 25, 98, 53, 60, 23, 100, 55, 58]



[46, 67, 30, 71, 48, 65, 28, 75, 50, 63, 26, 85, 52, 61, 24, 99, 54, 59, 22, 101]

Uplynutý čas: 0.0139923095703125 s

Mám úspešne odskúšanú aj veľkosť 31x31, ale samotný výstup sa nezmestí do wordu.