

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ
SLOVENSKÁ TECHNICKÁ UNIVERZITA
Ilkovičova 2, 842 16 Bratislava 4

2021/2022
Umelá inteligencia
Zadanie č.2 – Eulerov kôň

Cvičiaci: Ing. Ivan Kapustík
Čas cvičení: Štvrtok 12:00 – 13:50

Vypracovala: Monika Zjavková
AIŠ ID: 105345

Obsah

1. Zadanie – Problém 3. f).....	3
2. Opis riešenia	3
3. Zhodnotenie riešenia a testovanie.....	4

1. Zadanie – Problém 3. 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 východzích bodov (spolu teda 10 východzích 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.

2. Opis riešenia

Pri riešení je použitý algoritmus prehľadávania do hĺbky. Je to úplný algoritmus, ktorý vždy nájde a riešenie alebo dospeje k záveru, že riešenie neexistuje a skončí. Začína v políčku zadanom na začiatku a následne sa v algoritme rekurzívne prehľadávajú ostatné vrcholy. Postupuje ďalej podľa toho, ako mu to usporiadaná dvojica súradníc x a y dovoľia, aby imitovali pohyb koňa a stále sa nachádzal vo vnútri šachovnice. Kôň sa teda môže pohybovať podľa operátorov, kde prvé číslo znamená posun od aktuálnej pozície v riadku a druhé v stĺpci:

(1, 2), (1, -2), (2, 1), (2, -1),
(-1, 2), (-1, -2), (-2, 1) a (-2, -1).

Nájde vždy prvé políčko, v ktorom sú dodržané podmienky a zapíše číslo ťahu. Následne je rekurzívne znova hľadané ďalšie voľné políčko, ale tentokrát sa hýbe z toho, ktoré našlo v predchádzajúcom volaní. Ak prejde celý zoznam povolených pohybov, znamená to, že aktuálne šachovnica nemá riešenie a rekurzívna vetva sa ukončí po tom, ako vynuluje najnovší ťah.

Algoritmus skončí vtedy, keď už nie je možné sa pohnúť na žiadne iné políčko a všetky ostatné možnosti boli vyčerpané alebo je splnená podmienka, že počet ťahov sa rovná veľkosti šachovnice, čiže nájde prvé správne riešenie, ktoré sa vypíše aj s časom, ako dlho mu prehľadávanie trvalo.

Prehľadávanie je obmedzené časom a v prípade, že nestihne dospieť k riešeniu v požadovanom čase, prehľadávanie skončí a vypíše sa, že riešenie nebolo možné nájsť.

Pamäťová náročnosť je $O(n)$, kde n predstavuje počet políček na šachovnici. Časová zložitosť je v najhoršom prípade $O(8^n)$, kde n je tiež predstavuje počet políček.

Údaje o šachovnici sú reprezentované ako dvojrozmerné pole, ktoré je naplnené nulami a následne sa zapíňa poradovým číslom ťahu. Každá matica s číslami počas prehľadávania predstavuje jeden stav, v ktorom sa aktuálne nachádza. Správny finálny stav potom je matica, ktorá je naplnená číslami od jedna po počet políček na šachovnici, pričom každé číslo sa vyskytne práve raz.

3. Zhodnotenie riešenia a testovanie

Implementácia algoritmu prehľadávania do hĺbky dokáže nájsť riešenie do piatich sekúnd pre veľkosť 5x5 nájde zvyčajne najneskôr okolo piatej sekundy. Vo väčších šachovniciach program trvá dlhšie, kým nájde všetky riešenia.

Prehľadávanie by bolo možné zoptimalizovať z hľadiska prehľadávaného času napríklad zapamätávaním si pozícií, kde už bolo pokúšané sa vložiť číslo s daným ťahom, keďže pri rekurzii sa môžu prekrývať vo veľkých šachovniciach niektoré pokusy, čo by ušetrilo čas. Je to nevýhodnejšie však pre pamäťovú náročnosť, pretože by bolo potrebné ukladať. Ďalšia možnosť optimalizácie je použitie hashovania.

V šachovnici s veľkosťou 5x5 je možné riešenie z 13 políčok, presnejšie každé druhú políčko má riešenie začínajúc prvým na pozícii 0,0. Testovať správnosť riešenia je teda možné porovnaním pozície, či sa začínalo z jedného z riešiteľných. Riešenie všetkých políčok prebehlo správne v časovom limite.

Začiatočné pozície, kde existuje riešenie sú teda tieto:

pozície	0	1	2	3	4
0	Á	N	Á	N	Á
1	N	Á	N	Á	N
2	Á	N	Á	N	Á
3	N	Á	N	Á	N
4	Á	N	Á	N	Á

Potom pri testovacom scenári sa dá skontrolovať, či sú riešenia dobré podľa tabuľky vyššie. V tomto konkrétnom nenašlo riešenie pri bodoch 4,1 a 2,1; kde aj v tabuľke he uvedené, že žiadne riešenie neexistuje.

```

Bod č. 1
súradnice: [0, 4]
Čas: 0.2884790897369385
Riešenie: [[25, 14, 19, 8, 1], [6, 9, 2, 13, 18], [15, 24, 7, 20, 3], [10, 5, 22, 17, 12], [23, 16, 11, 4, 21]]

Bod č. 2
súradnice: [4, 0]
Čas: 0.00976109504699707
Riešenie: [[23, 20, 15, 8, 25], [14, 9, 24, 21, 16], [19, 22, 13, 4, 7], [10, 5, 2, 17, 12], [1, 18, 11, 6, 3]]

Bod č. 3
súradnice: [4, 1]
Čas: 6.6004626750946045
Riešenie: Riešenie sa nenašlo

Bod č. 4
súradnice: [2, 1]
Čas: 5.653823137283325
Riešenie: Riešenie sa nenašlo

Bod č. 5
súradnice: [1, 3]
Čas: 1.6138081550598145
Riešenie: [[23, 6, 11, 16, 25], [12, 17, 24, 1, 10], [7, 22, 5, 18, 15], [4, 13, 20, 9, 2], [21, 8, 3, 14, 19]]

```

Pri veľkosti šachovnice 6x6, je možné nájsť riešenie z každého políčka. Vzhľadom na to, že algoritmus nie je najviac efektívny a hľadá len prvú cestu, pri čom prehľadáva každé políčko, nestihne nájsť každé riešenie v časovom limite.

```
Bod č. 1
súradnice: [0, 5]
Čas: 15.0
Riešenie: Riešenie sa nenašlo

Bod č. 2
súradnice: [5, 2]
Čas: 4.973
Riešenie: [[36, 11, 20, 29, 34, 31], [19, 28, 35, 32, 13, 22], [10, 5, 12, 21, 30, 33], [27, 18, 3, 6, 23, 14], [4, 9, 16, 25, 2, 7], [17, 26, 1, 8, 15, 24]]

Bod č. 3
súradnice: [2, 2]
Čas: 15.0
Riešenie: Riešenie sa nenašlo

Bod č. 4
súradnice: [5, 4]
Čas: 0.252
Riešenie: [[36, 27, 14, 19, 30, 25], [13, 20, 29, 26, 15, 18], [28, 35, 12, 17, 24, 31], [21, 4, 23, 10, 7, 16], [34, 11, 2, 5, 32, 9], [3, 22, 33, 8, 1, 6]]

Bod č. 5
súradnice: [0, 3]
Čas: 15.0
Riešenie: Riešenie sa nenašlo
```

Najrýchlejšie našlo políčka v posledných dvoch riadkoch, kde sa v limite podarilo nájsť v štvrtom riadku 5 riešení a v piatom riadku našlo v časovom limite 4. Všeobecne by mali byť rýchlejšie rohové pozície, prípadne všetky krajné, kde kôň má menej možností na ktoré políčka sa posunie a tým sa skracuje počet možných rekurzívnych vetiev.