Zadanie 2 – Eulerov kôň

Umelá inteligencia

# Špecifikácia problému:

A screenshot of a computer

Description automatically generated

# Zadanie:

A close-up of a text

Description automatically generated

# Implementácia riešenia:

Implementáciu problému som sa rozhodol robiť v Pythone. Ako riešenie tohto zadania som použil klasické **prehľadávanie stromu do hĺbky (DFS)**. Implementácia zadania je pre šachovnice 5x5 a 6x6. Pri šachovniciach o veľkosti 7x7 je prehľadávaný strom príliš veľký a algoritmus časovo nestihne vyhľadať riešenie (po odstránení časového a krokového limitu trvalo viac ako hodinu, po hodine som program prerušil).

# Program časť 1

Spomínaný program som v tejto dokumentácii rozdelil na 3 časti. V tejto časti opíšem, čo sa s programom stane priamo po spustení programu.

Po spustení programu vybehne do Shellu provizórne menu, ktoré opisuje jednotlivé módy spustenia.

A screen shot of a computer

Description automatically generated

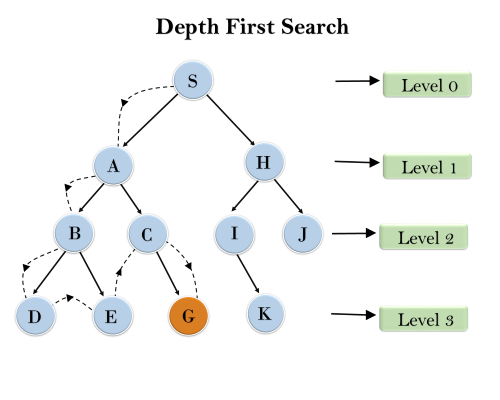
Prvý mód je „debugovací“ alebo pomocný mód, ktorým sa šikovne dá pozrieť, ako program funguje krok po kroku.

Ďalší mód je samostatné zadanie, kde sa vygeneruje 5 náhodných pozícií (z toho jedna je vždy ľavý dolný roh) pre šachovnice 5x5 a 6x6.

Tretí a posledný mód je len rýchle overenie riešenia pre zadaný problém.

# Program časť 2

V tejto časti vysvetlím algoritmus riešenia zadania, a to už spomínaný algoritmus prehľadávania stromu do hĺbky.



Zdroj: <https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.javatpoint.com%2Fai-uninformed-search-algorithms&psig=AOvVaw1pi17mthAr91JeVAO-k_gf&ust=1697188734826000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoTCOi3-u6W8IEDFQAAAAAdAAAAABAJ>

**Algoritmus DFS:**

1. Nastav poradové číslo ťahu
2. Vyhľadanie všetkých možných krokov pre koňa (child nodes)
3. Zober prvý vyhľadaný krok, nastav mu poradové číslo + 1 a rekurzívne zavolaj metódu
4. Pokiaľ vráti Int (indikácia True), sčíta počet krokov doposiaľ vykonaných s krokmi vykonanými pri mojej iterácii.
5. Pokiaľ vráti False, vráť sa a ohlás neúspešnosť.

**Obmedzenia algoritmu:**

Pre účely zadania boli implementované nasledovné limitery:

**Časový limiter:**

A screen shot of a computer

Description automatically generated

Časový limiter meria čas od spustenia algoritmu. Maximálny čas behu algoritmu je v premennej TIME\_LIMIT (teraz na stavený na 15 sekúnd).

**Krokový limiter:**

****

Kód...

**A black background with white text

Description automatically generated**

Tento limiter slúži na obmedzenie počtu krokov, ktoré môže algoritmus vykonať. Limit je nastavený v premennej MAX\_MOVES, a momentálne je to 50 000.

# Program časť 3

Táto časť opisuje poslednú fázu programu, a to je finálny výpis ktorý je vidieť po konci programu.

A screenshot of a computer

Description automatically generated

Program pri úspešnom vyhľadaní vypíše informácie o nájdenom riešení, ako štartovaciu pozíciu, čas alebo počet krokov. Pri neúspešnom riešení program vypíše, že vyhľadanie nebolo úspešné, a pokiaľ to bolo ukončenie pomocou jedného z limiterov, vypíše aj, že jeden z limitov bol prekročený.