**Umelá inteligencia Zadanie 1**

**Použite algoritmus lačného hľadania, porovnajte výsledky heuristík 1. a 2.**

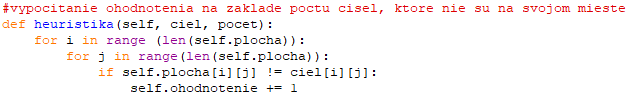
Marek Smutný

Riešenie problému som programoval v Pythone. Údaje (postupnosť krokov) som reprezentoval štruktúrou stromu. Jeden uzol obsahuje konkrétny stav plochy pomocou dvojrozmerného poľa, ohodnotenie, deti (možnosti, kam sa vieme posunúť) a rodiča (stav plochy, ktorý bol predtým).

Obrázok, na ktorom je text

Automaticky generovaný popis

Ohodnotenie počíta funkcia heuristika. Vždy musí byť odkomentovaný iba jeden typ! Prvý typ spočíta počet políčok, ktoré nie sú na mieste na ktorom majú byť vo finálnom stave.



Druhý typ spočíta vertikálne a horizontálne vzdialenosti každého čísla od miesta, na ktorom sa má nachádzať vo finálnom stave.

Obrázok, na ktorom je text

Automaticky generovaný popis

Funkcia vytvor\_strom rekurzívne vytvára deti uzlov s najmenším ohodnotením a kontroluje, či sa už nenachádzame vo finálnom stave (ohodnotenie = 0). Veľmi dôležitá podmienka v tejto funkcii, kontroluje, či uzly, ktoré vytvárame nemajú väčšie ohodnotenie ako uzly, ktoré už boli vytvorené ale neboli rozvinuté.

Obrázok, na ktorom je text

Automaticky generovaný popis

Využíva Priority queue uzly, v ktorej sú uložené všetky nerozvinuté uzly.

Obrázok, na ktorom je text

Automaticky generovaný popis

Všetky rozvinuté uzly sa ukladajú do hashovacej tabulky (tabulka) importoval som funkcie z modulu hash\_zoznam (tento súbor som vytvoril minulý semester na predmete DSA)



Obrázok, na ktorom je text

Automaticky generovaný popis

Po spustení programu je potrebné zadať veľkosť poľa a jednotlivé hodnoty v poli, pri každom inpute sú vysvetlivky.

**Testovanie**

Korektnosť môjho riešenia som kontroloval pomocou stránky <https://deniz.co/8-puzzle-solver/> . Vykonal som desiatky testov a moje riešenie sa vždy dostalo do cieľa (pri heuristike h(2)). Počet vytvorených uzlov sa v porovnaní s internetovou verziou vždy líšil. Niekedy bolo efektívnejšie riešenie na internete, inokedy zas moje riešenie.

Napríklad: [3, 1, 8]

[7, 2, 5]

['m', 4, 6]

Moje riešenie: 243 uzlov, Hĺbka 77

Internet: 384 uzlov, Hĺbka 52

['m', 2, 4]

[1, 7, 8]

[6, 5, 3]

Moje riešenie: 228 uzlov, Hĺbka 75

Internet: 226 uzlov, Hĺbka 22

[2, 5, 3]

[4, 1, 8]

['m', 6, 7]

Moje riešenie: 1027 uzlov, Hĺbka 141

Internet: 153 uzlov, Hĺbka 34

[5, 4, 7]

[1, 'm', 2]

[8, 6, 3]

Moje riešenie: 1300 uzlov, Hĺbka 173

Internet: 180 uzlov, Hĺbka 36

[1, 4, 8]

[2, 6, 3]

[7, 5, 'm']

Moje riešenie: 1045 uzlov, Hĺbka 161

Internet: 1184 uzlov, Hĺbka 62

**Porovnanie h(1) a h(2):**

V nasledujúcom teste som vkladal vstupy v stĺpci príklad (pri veľkosti 3x3).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **3x3** | **Pocet uzlov** | | **Dlzka cesty** | |
| **priklad** | **h1** | **h2** | **h1** | **h2** |
| **743106528** | **1570** | **420** | **273** | **59** |
| **516402738** | **1226** | **250** | **99** | **43** |
| **510342786** | **1315** | **278** | **105** | **57** |
| **135206784** | **1701** | **506** | **99** | **59** |
| **756423081** | **3555** | **543** | **329** | **111** |

V nasledujúcich testoch som vyberal najťažšie príklady pri rôznych veľkostiach na stránke <http://www2.fiit.stuba.sk/~kapustik/MN%20hlavolam.html> . Z výsledkov je zrejmé, že heuristika č.2 je efektívnejšia vo všetkých smeroch.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **najťažší príklad zo stránky (h1)** | **2x3** | **2x4** | **3x3** | **2x5** | **3x4** |
| **počet uzlov** | **61** | **1140** | **1076** | **5980** | **20066** |
| **dĺžka cesty** | **30** | **213** | **164** | **626** | **524** |
| **čas** | **0,01** | **0,06** | **0,07** | **0,88** | **11,32** |
| **najťažší príklad zo stránky (h2)** | **2x3** | **2x4** | **3x3** | **2x5** | **3x4** |
| **počet uzlov** | **70** | **257** | **155** | **560** | **2817** |
| **dĺžka cesty** | **38** | **85** | **58** | **142** | **288** |
| **čas** | **0,01** | **0,025** | **0,03** | **0,04** | **0,4** |