

Zadanie

V probléme 2 použiť A* algoritmus a následne porovnať výsledky heuristik 1. a 2.
Definovanie problému 2 - Našou úlohou je nájsť riešenie 8-hlavalamu. Hlavalam je zložený z 8 očíslovaných políček a jedného prázdneho miesta. Políčka je možné presúvať hore, dole, vľavo alebo vpravo, ale len ak je tým smerom medzera. Je vždy daná nejaká východisková a nejaká cieľová pozícia a je potrebné nájsť postupnosť krokov, ktoré vedú z jednej pozície do druhej.

Opis riešenia

Toto zadanie som riešil v jazyku python nasledujúcim spôsobom. Program je rozdelený do štyroch .py súborov. Súborov vysvetľujúc už v mene na čo budú v zadaní slúžiť.

Na začiatku sa na výstupe vypíšu vety, ktoré informujú používateľa o tom, aby si zvolil obťažnosť vstupu. Následne sa na výstup vypíšu ďalšie vety, ktoré používateľa nabádajú, aby si zvolil jednu z možností, ktorú konkrétnu heuristiku pre algoritmus a* chce v tom to zadaní použiť. V zadaní mám dve funkcie a*, v podstate robia to isté, len je v nich implementovaná iná heuristická funkcia. Jedna heuristická funkcia mi počíta počet políček, ktoré nie sú na svojom mieste a druhá heuristická funkcia mi vypočíta súčet vzdialeností jednotlivých políček od ich cieľovej pozície.

V oboch týchto a* funkciách si meriam aj čas za aký sa problém vyrieši, respektíve nevyrieši. V programe mám nastavený limit 15 sekúnd dokedy maximálne sa môže program vykonávať. V oboch funkciách a* mám zoznamy, konkrétne zoznam na uzly, ktorý sa vola nodes. Do zoznamu nodes budem pridávať pomocou funkcie append uzol, ktorý bude obsahovať aktuálny stav hlavalamu, aktuálny ťah, podľa heuristiky koľko políček nie je na svojom mieste respektíve sumu jednotlivých políček od ich cieľovej pozície, súčet aktuálneho ťahu a výsledok heuristickej funkcie a nakoniec bude obsahovať predchádzajúce rozloženie hlavalamu, čiže dokopy päť informácií bude mať jeden uzol. Zoznam na pohyby v hlavalame, ktoré som už spravil s názvom visited, aby som zbytočne neopakoval tie isté pohyby. Ďalší zoznam, ktorý používam, sa volá path. Do toho zoznamu pomocou funkcie append pridávam hlavalam, ktorý idem ďalej prehľadať, neskôr z tohoto zoznamu zisťujem kroky, ako som dospel k riešeniu.

Posuny v hlavalame riešim nasledujúco: ak sa nula nenachádza na indexe 2,5 alebo 8 tak viem, že sa môžem pohnúť doprava, ak sa nula nebude nachádzať na indexe 0,3 alebo 6, tak môžem ísť doľava, ak sa nula nenachádza na indexe 0, 1 alebo 2, tak sa môžem pohnúť hore, alebo posledná možnosť ak sa nula nenachádza na mieste 6,7 alebo 8, tak sa môžem posunúť dole. Voľné políčko mi predstavuje nula, teda čísla sa budú môcť v hlavalame vymeniť s nulou.

Ak som už vyskúšal všetky smery, tak pomocou for cyklu prejdem zoznam nodes či sa už tam nenachádza cieľový stav. Ak áno, vypíšem príslušné výpisy, že som našiel riešenie a ak nie, tak pokračujem v hľadaní. Zoznam nodes taktiež usporadúvam pomocou funkcie sorted a to na základe informácií o súčte aktuálneho pohybu plus výpočtu zvolenej heuristickej

funkcie. Ak som nenašiel riešenie, vyberiem z usporiadaného zoznamu hneď prvé teda najmenšie a opäť sa skúsim posunúť do vyhovujúcich smerov a tak ďalej sa to cyklí, kým nie je nájdené riešenie, respektíve uplynul čas 15 sekúnd.

Spôsob testovania

Zadanie testujem na štyroch rôznych vstupoch s názvom easy, medium, hard a unsolvable. Názvy už napovedajú náročnosť týchto testov. Spôsob akým to testujem je nasledujúci – zvolím si obťažnosť a spustím heuristickú funkciu jednotlivo alebo obe naraz. Ak spustím obe naraz, tak sa mi pri oboch vypíšu presné kroky ako som dospel k riešeniu, koľko krokov bolo potrebných na vyriešenie a za aký čas sa hlavolam vyriešil. Nastavený je časový limit 15 sekúnd do akého sa hlavolam musí vyriešiť inak sa považuje za neriešiteľný.