Slovenská technická univerzita v Bratislave Fakulta informatiky a informačných technológií

Riešenie 8 hlavolamu pomocou algoritmu A *

Martin Bopkó
zadanie 2

cvičenie: štvrtok 16:00
cvičiaci: Ing. Boris Slíž

1. Definovanie problému

Našou úlohou je nájsť riešenie 8-hlavolamu. Hlavolam je zložený z 8 očíslovaných políčok 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.

2. Opis riešenia

2.1

Mojou úlohou bolo riešiť zadanie pomocou algoritmu *A**. Z textového súboru *puzzle.txt* načítam cieľový a začiatočný stav. Súbor *ostatne test dvojice.txt* obsahuje ďalšie vstupy. Zo začiatočného stavu postupne vygenerujem potenciálne nové stavy (cesta k cieľu) pomocu rekurzívnej funkcie. Maximálne možem vytvoriť 4 nové stavy, keďže len do 4 smerov môžem posúvať čísla v matici. Na vygenerovanie nasledujúcih stavov používam dátovú štruktúru strom/*tree* . Každý nový stav dám aj do usporiadaného spájaného zoznamu (*open list*). Z tohoto zoznamu si vyberiem, že ktorý uzol/stav bude nasledujúci smer. Keď z uzla vytvorím všetky možné stavy/smery, tak rodičný uzol, ktorý vždy je na začiatku *open listu* vymažem zo zoznamu a pridám do druhého zoznamu (*closed list*), ktorý obsahuje všetky navštívené uzly. Takto postupujem, až kým sa nenájde cieľový stav, potom rekurzívne vypýšem cestu od konečného stavu.

2.2

Ošetril som zadanie podľa 2 heuristických hodnôt. 1. hodnota - ***Hamming Distance/Misplaced Tiles*** sa rovná s počtom políčok, ktoré nie sú na svojom mieste. Teoreticky táto hodnota je pomalšia ako druhá, a tiež používa viac uzlov na riešenie. 2. hodnotu - ***Manhattan Distance*** reprezentuje súčet vzdialeností jednotlivých políčok od ich cieľovej pozície. Napr bod1 má vzdialenosť od bodu dva danú formulou |x1-x2|+|y1-y2|.

Štruktúra stavu:

```
struct node
{
   char puzzle[3][3];
   struct node *left, *right, *up, *down, *parent, *next, *nextstack;
   int heuristicv;
   int depth;    char instance_of;
}*start = NULL;
```

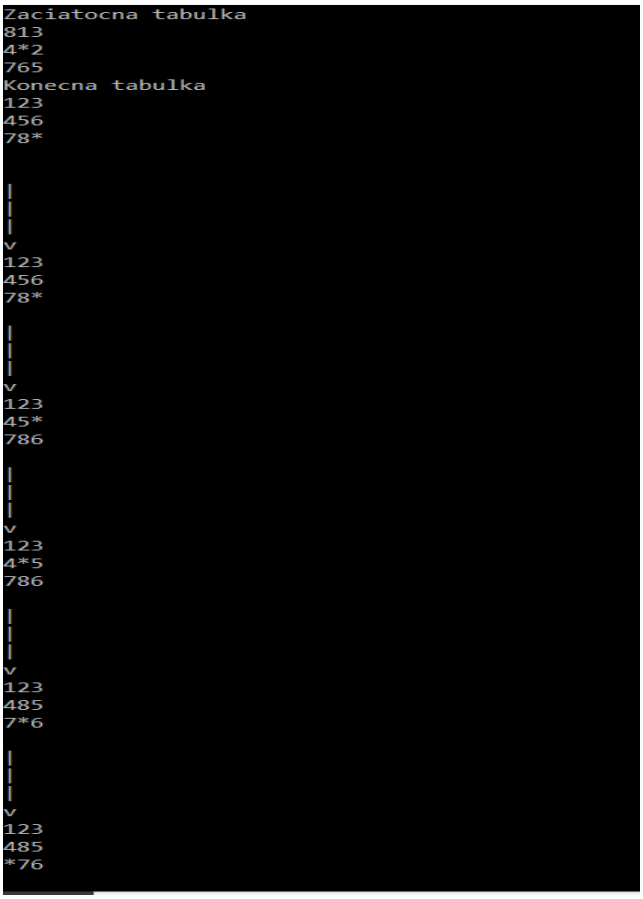
V 2d poli ukladám maticu čísel (prázdne políčko reprezentuje znak *. Ukazovatele *right, *left, *up, *down ukazujú na nové listy, podľa toho, že do ktorého semru sa posúva prázdna hodnota, teda znamienko *. *parent ukazuje späť na rodičný uzol. *next slúži na to, aby som ich vedel ukladať do *open listu*. Heuristicv sa rovná hodnotou heuristiky a instance_of reprezentuje jedno písmeno, ktoré mi určí, že v akom smere bol stav vytvorený. Inak by program vygeneroval aj predošlé rodičné stavy, čo može viecť k nekonečne veľa novým stavom.

3. Programovacie prostredie

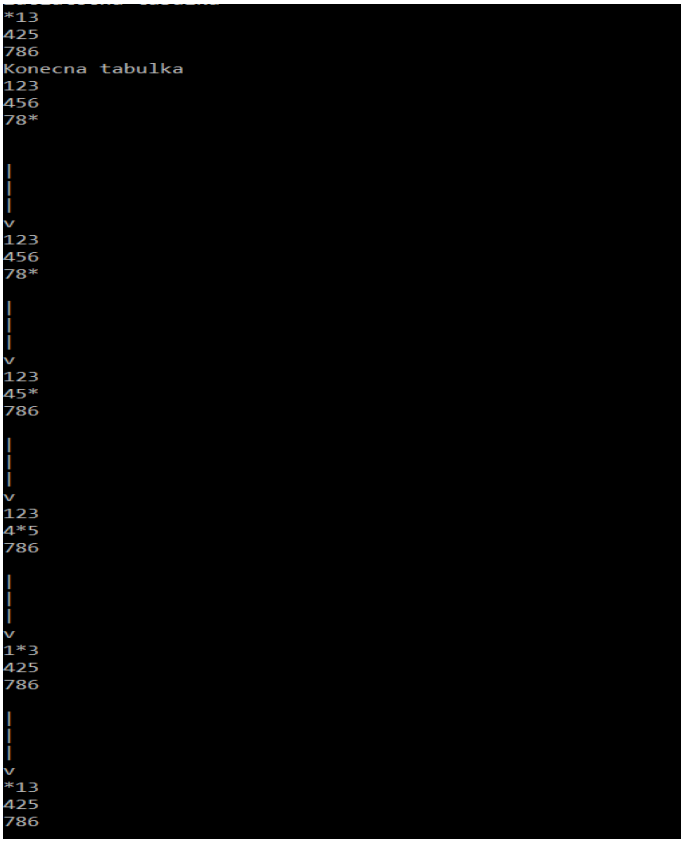
Úlohu som riešil v jazyku C, pomocu kompilátora CodeBlocks.

Výstupy na testované hlavolamy (vypísané zo zadu):

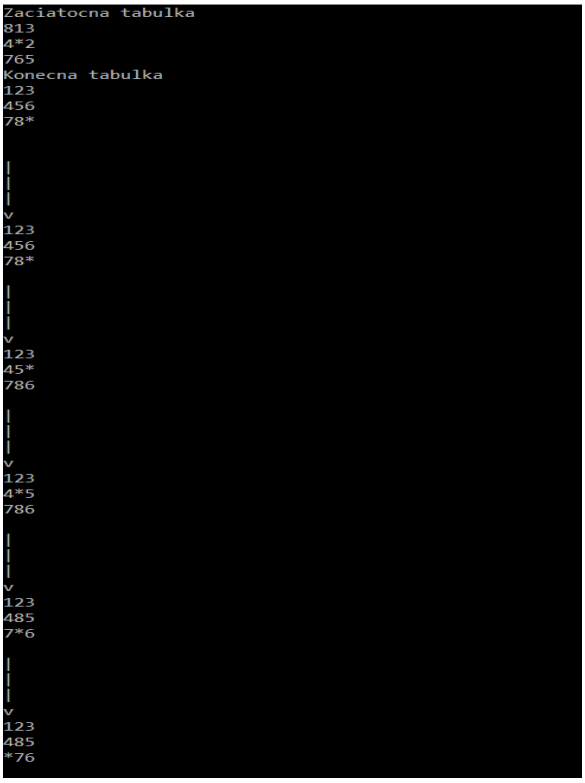
test č. 8



Test č. 7


*13
425
786
Konecna tabulka
123
456
78*
|
|
|
v
123
456
78*
|
|
|
v
123
45*
786
|
|
|
v
123
4*5
786
|
|
|
v
1*3
425
786
|
|
|
v
*13
425
786

Test č. 3


Zaciatocna tabulka
813
4*2
765
Konecna tabulka
123
456
78*
123
456
78*
123
45*
786
123
4*5
786
123
485
7*6
123
485
*76