FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ´

SLOVENSKÁ TECHNICKÁ UNIVERZITA

Ilkovičova 2, 842 16 Bratislava 4

2023/2024

Umelá Inteligencia

8 Hlavolam (Zadanie 1. d)

Učiteľ: Ing. Juraj Vincúr, PhD.

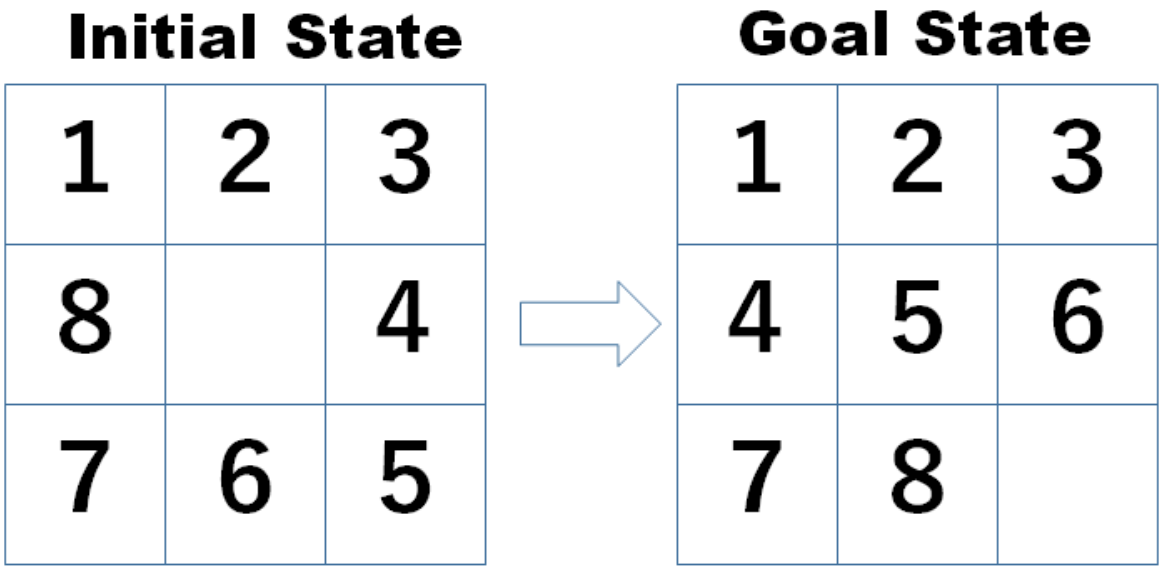
Čas: Streda 16:00 – 17:50

Realizované: Svätopluk Puterka

AIS ID: 120870

1 Úvod

Úloha spočíval v riešení 8-puzzle hlavolamu. Jedná sa o presunutie 8 čísel do správnych pozícií, pričom čísla vždy môžeme posúvať na miesto prázdneho políčka.



2. Použité Heuristické funkcie

Pri mojej implementácii som implementoval 2 heuristiky:

Manhattanskú vzdialenosť:

- daná heuristika počíta vzdialenosti medzi dvoma pozíciami, ktorá sa meria ako súčet horizontálnej a vertikálnej vzdialeností.
- celková Manhattanská vzdialenosť je suma všetkých týchto vzdialeností pre všetky čísla na hernom pláne
- moja implementácia

```python
def heuristic_manhattan_distance(game_plan: GamePlan) -> int:
    """
        Returns: cum of Manhattan Distances between goal state and
        current state of the game plan.
        Manhattan distance formula := |x1 - x2| + |y1 - y2|
        https://en.wikipedia.org/wiki/Taxicab_geometry
    """
    game_plan_side_length: int = get_side_length(game_plan)
    distance_sum = 0
    for index, value in enumerate(game_plan):
        if value == 0:
            continue
        # get position axis of a current value
        current_position = (
            index // game_plan_side_length,
            index % game_plan_side_length)
        # get goal position axis of the current value
        goal_position = (
            (value - 1) // game_plan_side_length,
            (value - 1) % game_plan_side_length)
        # calculate a distance between current and goal positions
        axis_x_difference = current_position[0] - goal_position[0]
        axis_y_difference = current_position[1] - goal_position[1]
        distance = abs(axis_x_difference) + abs(axis_y_difference)
        distance_sum += distance

    return distance_sum
```

Heuristiku nesprávne umiestnených políčok:

- daná heuristika počíta, koľko políčok nie je tam, kde by mali byť, vo výslednom riešení
- to znamená, že čím vyšší je jej výsledok, tým ďalej sme od riešenia.
- moje riešenie:

Moje riešenie spočíva v krokoch

1. kontrola či je problém riešiteľný

Ako prvé skontrolujem či je problém riešiteľný. Ak nie je riešitelný, tak ukončím program. Ak je vyriešiteľný tak pokračujem inicializáciou počiatočného stavu

```python
def is_solvable(game_plan: GamePlan) -> int:
    """
        verify if game is solvable
    """
    inversions = 0
    game_plan_length = len(game_plan)
    # get total number of inversions
    for compare_index in range(game_plan_length):
        for index in range(compare_index + 1, game_plan_length):
            if game_plan[compare_index] > game_plan[index] and game_plan[compare_index] != 0 and game_plan[index] != 0:
                inversions += 1
    return inversions % 2 == 0
```

2. inicializácia počiatočného stavu

Na začiatku som vložil iniciálny stav do prioritnej fronty. Stav pozostáva z aktuálneho usporiadania čísel na hracej ploche a historie krokov. ktorá je momentálne prázdna.

3. Konečný cyklus pozostávajúci z krokov

a. vybranie stavu s najnižšou prioritou z fronty
b. overíme či aktualne vybraný stav je cieľový stav
c. ak je stav cieľový, tak ukončujeme program
d. ak nie je stav cieľový pokračujeme ďalej
e. pre vybraný stav zistíme všetky možné stavy, ktoré môžu nastať po posunutí každého jedného suseda na prázdne políčko
f. pre každý z nových stavov
    i. skontrolujeme či už nie je v histórii stavov
    ii. ak stav nie je v histórii stavov, tak vypočítame zvolenou heuristickou funkciou jeho hodnotu a následne ho pridáme do prioritnej fronty aj s jeho indexom. Tento index nám neskôr pomôže zrekonštruovať postupnosť ťahov vedúcich k riešeniu

Porovnanie heuristických funkcii

Porovnanie oboch vyššie spomenutých heuristických funkcii.

Porovnanie prebehlo na rovnakých dátach. Z grafu možno vidieť, že vo väčšine prípadov bolo rýchlejšie riešenie, ktoré využívalo implementáciu heuristiky **Heuristika nesprávne umiestnených políčok**