**Zadanie 1: Piętnastka - instrukcja**

"Piętnastka", znana również pod angielską nazwą *"Fifteen Puzzle"*, jest układanką składającą się z ramki i osadzonych w niej 15 elementów. Elementy te można przesuwać, ponieważ w ramce pozostaje wolne pole o wielkości odpowiadającej jednemu elementowi (zatem cała ramka ma wymiar 4x4). Łamigłówka polega na takim przesuwaniu elementów układanki, aby z pewnego losowego układu początkowego, na przykład:

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | 2 | 7 |
| 8 | 9 | 12 | 10 |
| 13 | 3 | 6 | 4 |
| 15 | 14 | 11 | 5 |

uzyskać układ wzorcowy odpowiadający poniższemu:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 |  |

**Cel zadania**

Zadanie składa się z dwóch części: programistycznej i badawczej.

Celem części programistycznej jest napisanie programu, który będzie rozwiązywał powyższą łamigłówkę przy użyciu różnych metod przeszukiwania przestrzeni stanów:

* strategii "wszerz";
* strategii "w głąb";
* strategii "najpierw najlepszy": A\*, z następującymi heurystykami:
  + metryką Hamminga;
  + metryką Manhattan.

Cel części badawczej stanowi przebadanie, jak powyższe metody przeszukiwania przestrzeni stanów zachowują się w przypadku tego problemu.

**Część programistyczna**

**Wymagania funkcjonalne**

Program ma za zadanie generować rozwiązanie powyższej łamigłówki, czyli wyznaczać taki ciąg ruchów odpowiadających przesunięciom wolnego pola, które przeprowadzą układankę z zadanego układu początkowego do układu wzorcowego. Wyznaczony ciąg ruchów reprezentowany będzie za pomocą ciągu liter pochodzących z czteroelementowego zbioru, oznaczających poszczególne przesunięcia zgodnie z poniższą tabelą:

|  |  |
| --- | --- |
| **Przesunięcie** | **Litera** |
| w lewo (*left*) | L |
| w prawo (*right*) | R |
| w górę (*up*) | U |
| w dół (*down*) | D |

Ponadto program ma generować dodatkowe informacje dotyczące przeprowadzonego procesu obliczeniowego, obejmujące:

* długość znalezionego rozwiązania;
* liczbę stanów odwiedzonych;
* liczbę stanów przetworzonych;
* maksymalną osiągniętą głębokość rekursji;
* czas trwania procesu obliczeniowego.

Każda z metod przeszukiwania przestrzeni stanów wymaga podania dodatkowego parametru. W przypadku strategii "wszerz" i strategii "w głąb" jest to porządek przeszukiwania sąsiedztwa bieżącego stanu, który określony będzie za pomocą ciągu stanowiącego permutację czterech liter zamieszczonych w tabeli powyżej. Natomiast w przypadku strategii A\* jest to używana heurystyka.

Dodatkowo w przypadku strategii "w głąb" należy ustalić maksymalną dozwoloną głębokość rekursji, która nie może być mniejsza niż 20 (wartość tę można wpisać na sztywno do programu). W sytuacji, gdy program osiągnie taką głębokość nie znalazłszy rozwiązania, powinien wykonać nawrót.

Program będzie pobierać zadany układ początkowy układanki z pliku tekstowego o formacie opisanym poniżej (*Plik z układem początkowym*). Znalezione rozwiązanie program będzie zapisywać w innym pliku tekstowym o formacie opisanym poniżej (*Plik z rozwiązaniem*). Dodatkowe informacje dotyczące przeprowadzonego procesu obliczeniowego program również będzie zapisywać w osobnym pliku tekstowym o formacie opisanym poniżej (*Plik z dodatkowymi informacjami*).

Do programu będą przekazywane następujące argumenty wywołania:

* akronim określający wybraną strategię zgodnie z poniższą tabelą:

|  |  |
| --- | --- |
| **Strategia** | **Akronim** |
| "wszerz" (*breadth-first search*) | bfs |
| "w głąb" (*depth-first search*) | dfs |
| A\* (*A-star*) | astr |

* dodatkowy parametr wybranej strategii zgodnie z poniższą tabelą:

|  |  |
| --- | --- |
| **Parametr** | **Akronim** |
| porządek przeszukiwania sąsiedztwa | permutacja liter: L, R, U, D |
| heurystyka - metryka Hamminga | hamm |
| heurystyka - metryka Manhattan | manh |

* nazwa pliku tekstowego z zadanym układem początkowym układanki;
* nazwa pliku tekstowego, w którym ma zostać zapisane rozwiązanie;
* nazwa pliku tekstowego, w którym mają zostać zapisane dodatkowe informacje dotyczące przeprowadzonego procesu obliczeniowego.

Przykładowe wywołania programu:

* strategia "wszerz" z porządkiem przeszukiwania sąsiedztwa *prawo-dół-góra-lewo*:  
  program bfs RDUL 4x4\_01\_0001.txt 4x4\_01\_0001\_bfs\_rdul\_sol.txt 4x4\_01\_0001\_bfs\_rdul\_stats.txt
* strategia "w głąb" z porządkiem przeszukiwania sąsiedztwa *lewo-góra-dół-prawo*:  
  program dfs LUDR 4x4\_01\_0001.txt 4x4\_01\_0001\_dfs\_ludr\_sol.txt 4x4\_01\_0001\_dfs\_ludr\_stats.txt
* strategia A\* z heurystyką w postaci metryki Manhattan:  
  program astr manh 4x4\_01\_0001.txt 4x4\_01\_0001\_astr\_manh\_sol.txt 4x4\_01\_0001\_astr\_manh\_stats.txt

Wprawdzie program będzie służył do przebadania zachowania poszczególnych metod przeszukiwania przestrzeni stanów w przypadku klasycznej "Piętnastki", musi on być jednak napisany w sposób uniwersalny, to znaczy taki, który umożliwi generowanie rozwiązań także dla ramek o innych niż standardowe rozmiarach, w tym ramek niekwadratowych.

**Plik z układem początkowym**

Jest to plik tekstowy, w którym liczba linii zależy od rozmiaru ramki. Pierwsza linia zawiera dwie liczby całkowite *w* oraz *k*, oddzielone od siebie spacją, które określają odpowiednio pionowy (liczbę wierszy) i poziomy (liczbę kolumn) rozmiar ramki. Każda z pozostałych *w* linii zawiera *k* oddzielonych spacjami liczb całkowitych, które opisują położenie poszczególnych elementów układanki, przy czym wartość 0 oznacza wolne pole.

**Plik z rozwiązaniem**

Jest to plik tekstowy standardowo składający się z 2 linii. Pierwsza z nich zawiera liczbę całkowitą *n*, określającą długość znalezionego rozwiązania (czyli długość ciągu ruchów odpowiadających przesunięciom wolnego pola, które przeprowadzą układankę z zadanego układu początkowego do układu wzorcowego). Natomiast w drugiej linii znajduje się ciąg *n* liter odpowiadających poszczególnym ruchom wolnego pola w ramach znalezionego rozwiązania, zgodnie z reprezentacją przedstawioną w tabeli zamieszczonej wyżej. Jeżeli dla zadanego układu początkowego program nie znalazł rozwiązania, wówczas plik składa się tylko z 1 linii, która zawiera liczbę -1.

**Plik z dodatkowymi informacjami**

Jest to plik tekstowy składający się z 5 linii, z których każda zawiera jedną liczbę oznaczającą odpowiednio:

* 1 linia (liczba całkowita): długość znalezionego rozwiązania - o takiej samej wartości jak w pliku z rozwiązaniem (przy czym gdy program nie znalazł rozwiązania, wartość ta to -1);
* 2 linia (liczba całkowita): liczbę stanów odwiedzonych;
* 3 linia (liczba całkowita): liczbę stanów przetworzonych;
* 4 linia (liczba całkowita): maksymalną osiągniętą głębokość rekursji;
* 5 linia (liczba rzeczywista z dokładnością do 3 miejsc po przecinku): czas trwania procesu obliczeniowego w milisekundach.

**Część badawcza**

Należy przebadać wszystkie układy początkowe układanki w odległościach 1-7 od układu wzorcowego (w sumie 413 układów). W przypadku strategii "wszerz" i strategii "w głąb" użyć 8 następujących porządków przeszukiwania sąsiedztwa:

* *prawo-dół-góra-lewo*;
* *prawo-dół-lewo-góra*;
* *dół-prawo-góra-lewo*;
* *dół-prawo-lewo-góra*.
* *lewo-góra-dół-prawo*.
* *lewo-góra-prawo-dół*;
* *góra-lewo-dół-prawo*;
* *góra-lewo-prawo-dół*;

W przypadku strategii A\* użyć obu heurystyk. Korzystając z informacji zawartych w plikach z dodatkowymi informacjami dotyczącymi przeprowadzonego procesu obliczeniowego porównać skuteczność poszczególnych metod przeszukiwania przestrzeni stanów. Dane dotyczące poszczególnych aspektów rozważanych przy porównaniach zobrazować na wykresach (prezentacja danych w inny sposób skutkować będzie obniżeniem oceny). Wyciągnąć wnioski na podstawie zaprezentowanych wyników.

Ostatnia modyfikacja: wtorek, 21 lutego 2017, 15:10