

Data oddania: _____

Ocena: _____

Maciej Pracucik 216869

Adam Jóźwiak 216786

Zadanie 1: Piętnastka

1. Cel

Celem zadania było napisanie programu rozwiązującego zagadkę logiczną, Piętnastkę, poprzez wykorzystanie różnych metod przeszukiwania stanu oraz przebadanie ich.

2. Wprowadzenie

Piętnastka to układanka logiczna składająca się z piętnastu klocków, numerowanych od 1 do 15, ułożonych na kwadratowej planszy, o wymiarach 4x4. Celem układanki jest przestawienie tak klocków, żeby ustawić je w kolejności rosnącej, przy czym pusty element musi znaleźć się na końcu. Przesuwanie klocków umożliwia nam puste miejsce na planszy.

Poszukiwanie rozwiązania łamigłówki jest zbliżone do znajdowania ścieżki w grafie, gdzie stan układanki jest węzłem w grafu.

W celu odnalezienia właściwej ścieżki stosujemy następujące strategie:

- BFS - Breadth First Search - przeszukiwanie "wszerz" rozpoczynając od zadanego wierzchołka w grafie, odwiedza wszystkie osiągalne z niego wierzchołki na tym samym poziomie rekursji, według ustalonej wcześniej kolejności sprawdzania. Następnie odwiedza wszystkie osiągalne wierzchołki pochodne, na kolejnym poziomie rekursji
- DFS - Depth First Search - przeszukiwanie "w głąb" rozpoczyna przechodzenie grafu od zadanego wierzchołka, odwiedzając pierwszy z pochodnych wierzchołków, kolejność przechodzenia jest wcześniej ustalana,

powtarzając to dla każdego pochodnego wierzchołka. Jeżeli algorytm nie będzie mógł wchodzić dalej (osiągnie zadaną maksymalną głębokość rekursji lub już dalej nie będzie mógł wykonać ruchu), cofa się o jeden poziom rekursji i bada krawędź kolejnego nieodwiedzonego jeszcze wierzchołka.

- A* - Algorytm heurystyczny znajduje najkrótszą możliwą ścieżkę, jeśli taka istnieje. W przypadku piętnastki algorytm A* tworzy ścieżkę wybierając wierzchołek tak, aby minimalizować wartość heurystyki. Metody obliczania tej wartości to: metoda Hamminga, gdzie obliczamy ile klocków znajduje się na niewłaściwych pozycjach, metoda Manhattan, gdzie liczymy jakie odległości dzielą klocki od ich docelowych miejsc.

3. Opis implementacji

Program został napisany w języku Python. Klasa Fifteen jest klasą główną natomiast plik FileManager.py odpowiada za wczytywanie układanki, zapis statystyk oraz zapis ułożonej układanki. W klasie Fifteen znajdują się wszystkie strategie oraz niezbędne operacje na układance, np. odnalezienie pozycji pustej, zamiana elementów, przedstawienie możliwych ruchów. Konstruktor klasy Fifteen został tak przystosowany by pobierał parametry z PowerShella, przekazywane przez uruchamiacz, udostępniony na platformie Wikamp.

4. Materiały i metody

Algorytm BFS został napisany metodą rekurencyjną, gdzie na początek jest ustalana obecna głębokość równa 1, a następnie program próbuje ruszyć się w kierunku podyktowanym z góry. Jeśli mu się powiedzie, to metoda rekurencyjna wywołuje się ponownie, tym razem z wartością obecnej głębokości mniejszą o 1. Jeśli na najmniejszej głębokości równej 0 układanka będzie ukończona, program przestaje się wykonywać. W przeciwnym wypadku jest zwracany fałsz, po którym na poziomie wyższym wykonywany jest ruch w tył. Jeśli po wykonaniu każdego z dostępnych ruchów piętnastka nie będzie ułożona, to zwiększany jest poziom głębokości i program wykonuje się ponownie, do skutku.

Algorytm DFS został napisany metodą iteracyjną, gdzie każdy poziom rekursji, którego górną granicę ustawiliśmy na 20, przechowuje informację o ilości przebytych na nim możliwości. Na każdy poziom możliwe jest wykonanie maksymalnie 4 ruchów, w zależności od pozycji pustego elementu. Program schodzi na najniższy poziom uwzględniając wcześniej zadaną mu kolejność, a następnie przeszukuje tam wszystkie możliwości, sprawdzając czy układanka została ukończona. W przypadku niepowodzenia cofa się i przeszukuje kolejne odnogi poziomu na który się cofnął, do czasu aż wszystkie możliwe ruchy na danym poziomie zostaną wyczerpane.

Algorytm A* działa na zasadzie kolejki, z której wyłuskiwany jest element o najniższej wartości. Na początku program, w zależności od wybranej heurystyki oblicza który z możliwych do wykonania ruchów jest najbardziej

korzystny. W przypadku Metryki Hamminga, jest to suma elementów znajdujących się na nie swoich miejscach. Im suma jest mniejsza, tym bliżej do udanego wykonania. Zasada działania Metryki Manhattan jest bardzo podobna, również jest to suma, ale odległości każdego punktu, od jego docelowej pozycji. W przypadku gdy suma będzie równa 0, program zakończy działanie z sukcesem.

Badanie zostało przeprowadzone przy pomocy skryptu uruchamianego w Windows PowerShell. Skrypt pobierał jako argumenty rodzaj algorytmu, a także kolejność wykonywania ruchów. Następnie jego zadaniem było odczytanie wszystkich, wcześniej wygenerowanych plików .txt z początkowym układem piętnastki. Plik FileManager.py odpowiedzialny był w naszym programie za utworzenie odpowiednich plików ze statystykami danego przebiegu programu. Na koniec wszystkie osobno utworzone pliki przetworzone zostały za pomocą osobnego skryptu PowerShell'owego, a następnie zaimportowane do programu Microsoft Excel w celu dokonania analizy danych.

5. Wyniki

BFS

5.1. DRUL

BFS Uśrednione - DRUL					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]
1	1,000	1,000	1,000	1,000	0,000
2	10,500	6,250	5,000	2,000	0,250
3	41,400	22,200	14,900	3,000	1,402
4	115,583	59,792	36,083	4,000	4,044
5	283,074	144,037	82,278	5,000	9,490
6	693,738	349,869	191,776	6,000	22,824
7	1616,368	811,684	430,259	7,000	52,164

5.1.1. RDUL

BFS Uśrednione - RDUL					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]
1	1,000	1,000	1,000	1,000	0,000
2	10,500	6,250	5,000	2,000	0,250
3	41,400	22,200	14,900	3,000	1,402
4	115,583	59,792	36,083	4,000	4,044
5	283,074	144,037	82,278	5,000	9,490
6	693,607	349,869	191,776	6,000	22,824
7	1615,925	811,462	430,179	7,000	52,254

5.1.2. RDLU

BFS Uśrednione - RDLU					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]
1	1,000	1,000	1,000	1,000	0,000
2	10,500	6,250	5,000	2,000	0,250
3	41,400	22,200	14,900	3,000	1,401
4	115,583	59,792	36,083	4,000	3,921
5	283,074	144,037	82,278	5,000	9,518
6	693,738	349,869	191,776	6,000	22,855
7	1616,368	811,684	430,259	7,000	52,230

5.1.3. DRLU

BFS Uśrednione - DRLU					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]
1	1,000	1,000	1,000	1,000	0,000
2	10,500	6,250	5,000	2,000	0,510
3	41,400	22,200	14,900	3,000	1,602
4	115,583	59,792	36,083	4,000	4,003
5	283,074	144,037	82,278	5,000	9,638
6	693,607	349,804	191,766	6,000	22,978
7	1615,925	811,462	430,179	7,000	52,325

5.1.4. LUDR

BFS Uśrednione - LUDR					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]
1	5,000	3,000	3,000	1,000	0,000
2	26,500	14,250	11,000	2,000	0,501
3	74,200	38,600	25,700	3,000	2,602
4	162,250	83,125 50,417	4,000	5,296	
5	327,815	166,407	96,056	5,000	10,862
6	688,897	347,449	192,047	6,000	22,621
7	1517,925	762,462	406,594	7,000	49,877

5.1.5. LURD

BFS Uśrednione - LURD					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]
1	5,000	3,000	3,000	1,000	0,501
2	26,500	14,250	11,000	2,000	1,252
3	74,200	38,600	25,700	3,000	2,641
4	162,250	83,125	50,417	4,000	5,359
5	327,815	166,407	96,056	5,000	10,880
6	689,028	347,514	192,056	6,000	23,201
7	1518,368	762,684	406,675	7,000	49,777

5.1.6. ULDR

BFS Uśrednione - ULDR					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]
1	5,000	3,000	3,000	1,000	0,000
2	26,500	14,250	11,000	2,000	1,001
3	74,200	38,600	25,700	3,000	2,603
4	162,250	83,125 50,417	4,000	5,384	
5	327,815	166,407	96,056	5,000	10,907
6	689,028	347,514	192,056	6,000	22,818
7	1518,368	762,684	406,675	7,000	50,407

5.1.7. ULRD

BFS Uśrednione - ULRD					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]
1	5,000	3,000	3,000	1,000	0,000
2	26,500	14,250	11,000	2,000	1,001
3	74,200	38,600	25,700	3,000	2,502
4	162,250	83,125 50,417	4,000	5,463	
5	327,815	166,407	96,056	5,000	10,858
6	688,897	347,449	192,047	6,000	22,795
7	1517,925	762,462	406,594	7,000	50,067

DFS

5.2. DRUL

DFS Uśrednione - DRUL					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]

5.2.1. RDUL

DFS Uśrednione - RDUL					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]

5.2.2. RDLU

DFS Uśrednione - RDLU					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]

5.2.3. DRLU

DFS Uśrednione - DRLU					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]

5.2.4. LUDR

DFS Uśrednione - LUDR					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]

5.2.5. LURD

DFS Uśrednione - LURD					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]

5.2.6. ULDR

DFS Uśrednione - ULDR					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]

5.2.7. ULRD

DFS Uśrednione - ULRD					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]

A*

5.3. Metryka Hamminga

A* Uśrednione - Hamming					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]
1	7,000	1,000	1,000	1,000	0,000
2	13,000	2,000	2,000	2,000	0,501
3	19,000	3,000	3,000	3,000	0,500
4	28,185	4,167	4,167	4,167	0,959
5	28,185	394,870	9,778	394,870	75,100
6	29,290	994,084	15,514	994,084	188,031
7	27,123	1773,250	23,311	1773,250	340,677

5.4. Metryka Manhattan

A* Uśrednione - Manhattan					
Odległość od stanu zero	Długość rozwiązania	Liczba stanów odwiedzonych	Liczba stanów przetworzonych	Maksymalna głębokość rekursji	Czas wykonania [ms]
1	7,000	1,000	1,000	1,000	1,008
2	13,000	2,000	2,000	2,000	0,501
3	19,000	3,000	3,000	3,000	0,701
4	25,167	4,000	4,000	4,000	0,948
5	33,074	91,963	5,852	91,963	19,920
6	40,290	222,729	8,720	222,729	48,376
7	42,406	708,175	13,151	708,175	152,748

6. Dyskusja

Dla algorytmu BFS średnia ilość rozwiązań, przy niższych głębokościach jest mniejsza od A*, lecz przy wyższych to A* charakteryzuje się mniejszą ilością rozwiązań. Przykładowo maksymalna średnia ilość rozwiązań dla głębokości 7 dla algorytmu BFS jest to 1616,368 z kolei dla A* jest to zaledwie 42,406.

Ma to również wpływ na liczbę stanów odwiedzonych i przetworzonych, A* ma ich mniej od BFS, szczególne rozbieżności możemy zaobserwować dla

głębokości od 2 do 4. W A^* wartości rosną o 1 gdzie w BFS maksymalna różnica między głębokościami wynosi 30, przy większych głębokościach liczba stanów dla tych algorytmów jest stosunkowo podobna.

Maksymalna osiągnięta głębokość rekursji dla BFS rośnie stale o 1 dla kolejnych głębokości, dla A^* , dzieje się podobnie tylko na początku, ponieważ od głębokości 5 te wartości rosną o większą ilość, przy głębokości 7 dla metryki hamminga wynosi ona aż 1773,250.

Czasy wykonywania dla mniejszych głębokości są mniejsze w A^* , jednak dla wyższych to BFS ma mniejsze wartości, przy 7 głębokości metryka Hamminga średnio przechodziła przez 340ms, a BFS przechodził przez średnio 50ms. Dla algorytmu A^* wpływ miało również dobór heurystyki, metryka Hamminga potrafiła wykonywać się ponad 2 razy dłużej niż metryka Manhattan. Czas wykonywania algorytmu DFS jest znacznie dłuższy od pozostałych algorytmów, czyni go to bardzo niewygodnym algorytmem do odpowiedniego badania. Dodatkowy wpływ na czas wykonywania obliczeń mają różne czynniki takie jak, programy pracujące w tle, sposób przydzielania zasobów, czy też sama moc obliczeniowa naszego sprzętu.

7. Wnioski

- Do jak najszybszego ułożenia układanki najlepiej użyć algorytmu BFS lub A^* , ponieważ oboje charakteryzują się szybkim czasem wykonywania. Czas obliczeń algorytmu DFS jest zbyt długi by warto było go do tego zastosować.
- Heurystyka Manhattan nadaje się do bardziej złożonych układanek. Do prostszych nie ma to większego znaczenia.
- DFS charakteryzuje się zbyt dużą przypadkowością w znajdowaniu najkrótszego rozwiązania i pochłaniania zbyt dużo zasobów programowych. Ogromną wagę ma również dobranie odpowiedniego sposobu w jaki algorytm ma przeszukiwać układankę.

Literatura

- [1] T. Oetiker, H. Partl, I. Hyna, E. Schlegl. *Nie za krótkie wprowadzenie do systemu \LaTeX* 2e, 2007, dostępny online.
- [2] mgr Jerzy Wałaszek https://eduinf.waw.pl/inf/alg/001_search/0125.php
- [3] Joanna Raczyńska *Algorytm A^** <https://elektron.elka.pw.edu.pl/~jarabas/ALHE/notatki3.pdf>
- [4] Michał Karpiński <http://informatyka.wroc.pl/node/709>