

Sztuczna Inteligencja i Systemy Ekspertowe

Zadanie: Piętnastka

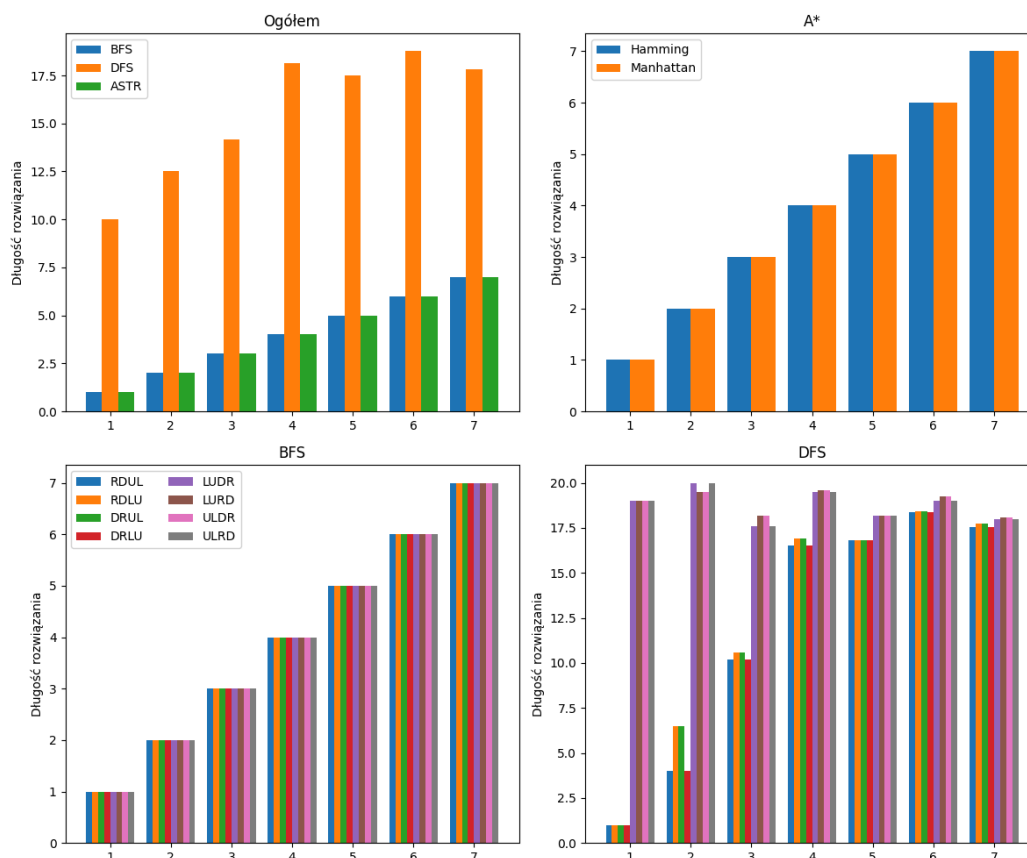
1. Cel

Napisać program rozwiązujący układankę “Piętnastkę” wykorzystujący strategię przeszukiwania “wszerz” (BFS), “w głąb” (DFS) i “najpierw najlepszy” (A* korzystając z metryki Hamminga i Manhattan).

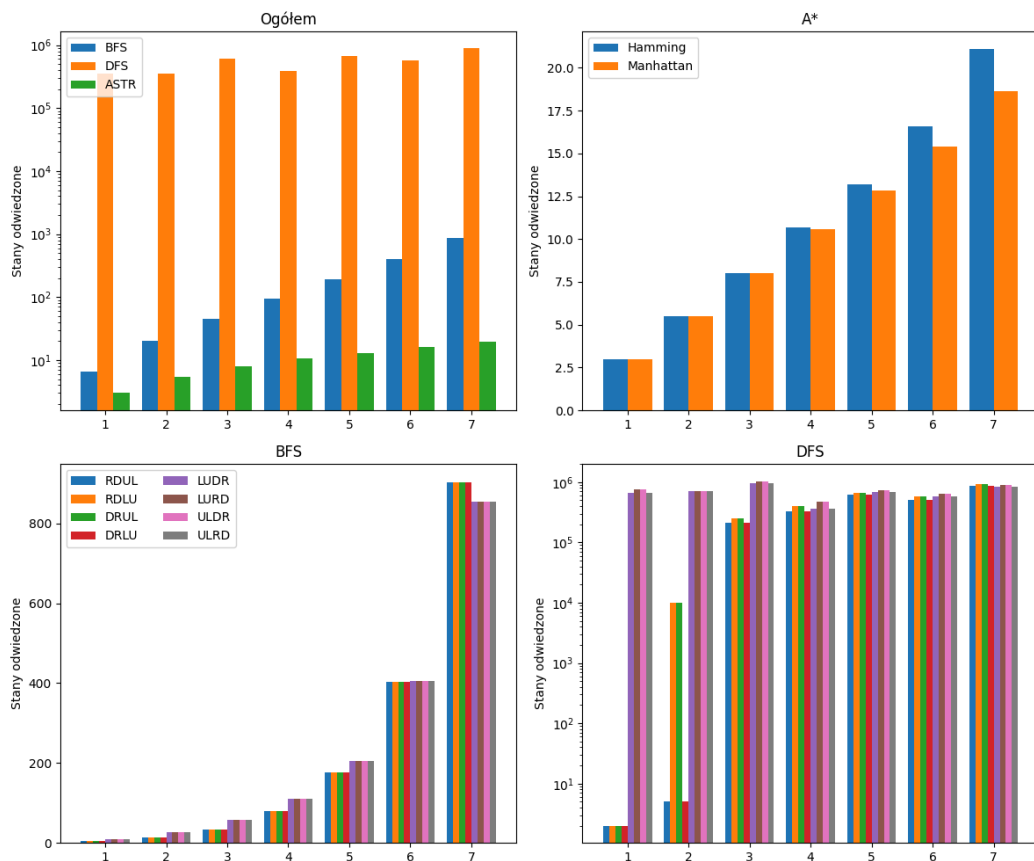
Na podstawie otrzymanych wyników ocenić korzyści stosowania poszczególnego algorytmu dla danych kryteriów (długość znalezionego rozwiązania, liczbę stanów odwiedzonych, przetworzonych, maksymalną osiągniętą głębokość rekursji i czas trwania procesu obliczeniowego).

2. Wyniki

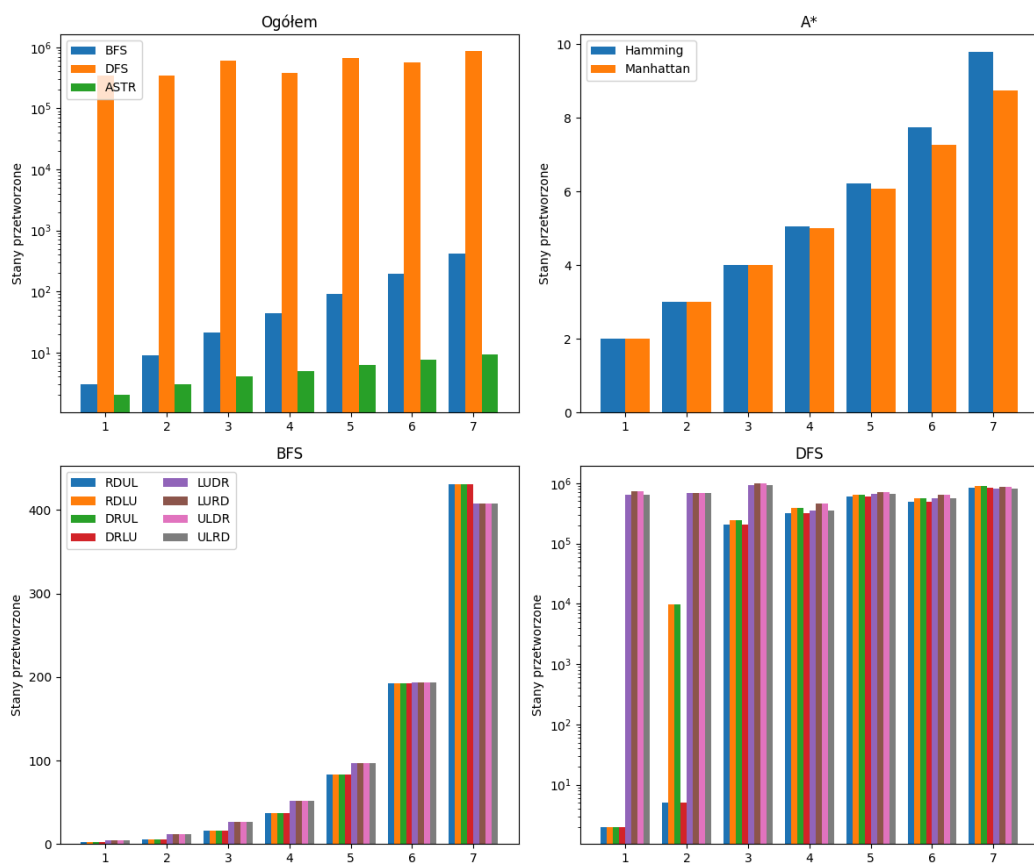
W celu realizacji zadania zaimplementowaliśmy program w języku Python z wykorzystaniem odpowiednich bibliotek, tj. Collections, Heapq, Matplotlib oraz Numpy.



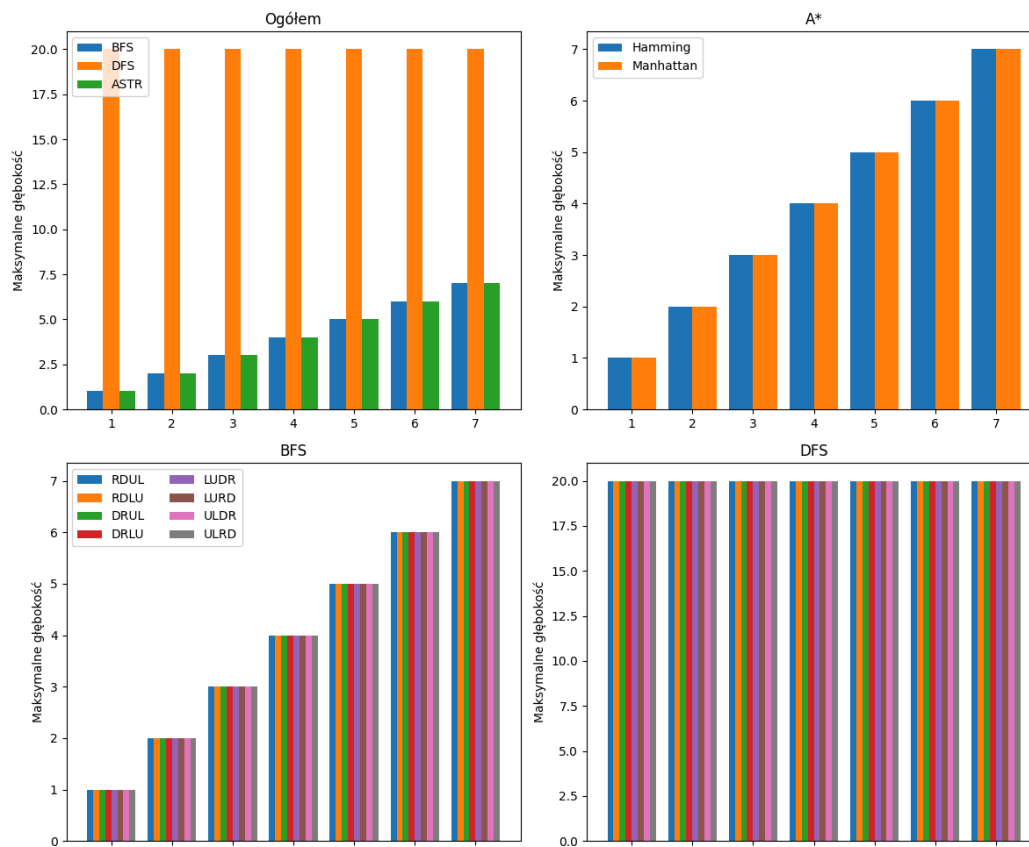
Rys. 1 Wykres charakterystyki średniej arytmetycznej długości rozwiązania względem głębokości rozwiązania



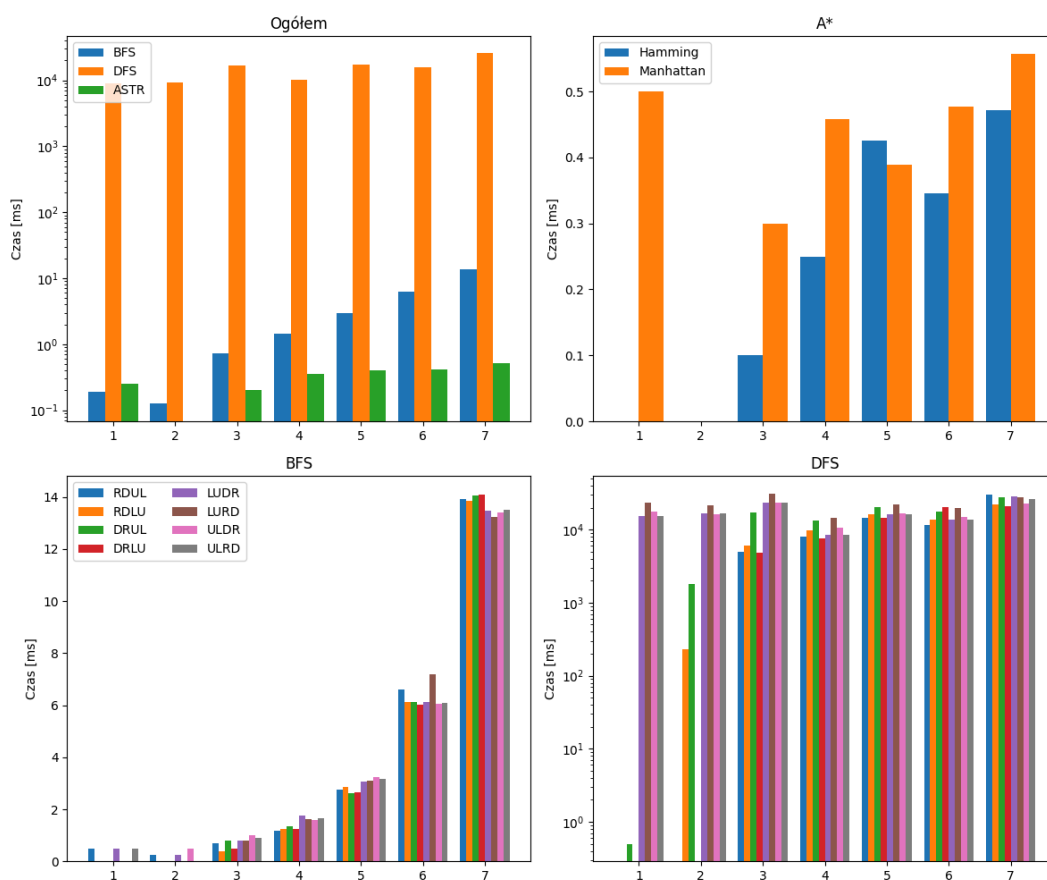
Rys. 2 Wykres charakterystyki średniej arytmetycznej liczby odwiedzonych stanów względem głębokości rozwiązania.



Rys. 3 Wykres charakterystyki średniej arytmetycznej liczby przetworzonych stanów względem głębokości rozwiązania



Rys. 4 Wykres charakterystyki średniej arytmetycznej maksymalnej głębokości względem głębokości rozwiązania.



Rys. 5 Wykres charakterystyki średniej arytmetycznej czasu trwania procesu obliczeniowego w milisekundach względem głębokości rozwiązania.

3. Wnioski

Analizując otrzymane wyniki widzimy, że algorytm A* najlepiej radzi sobie z rozwiązywaniem układanki “piętnastki”, natomiast DFS najgorzej, ponieważ ma tendencję do przekraczania maksymalnej głębokości rekurencji, co prowadzi do wykonania nawrotu. Liczba odwiedzonych i przetworzonych stanów oraz czas w algorytmie BFS rośnie eksponencjalnie wraz z odległością od układu wzorcowego.