

# Sztuczna inteligencja i systemy ekspertowe

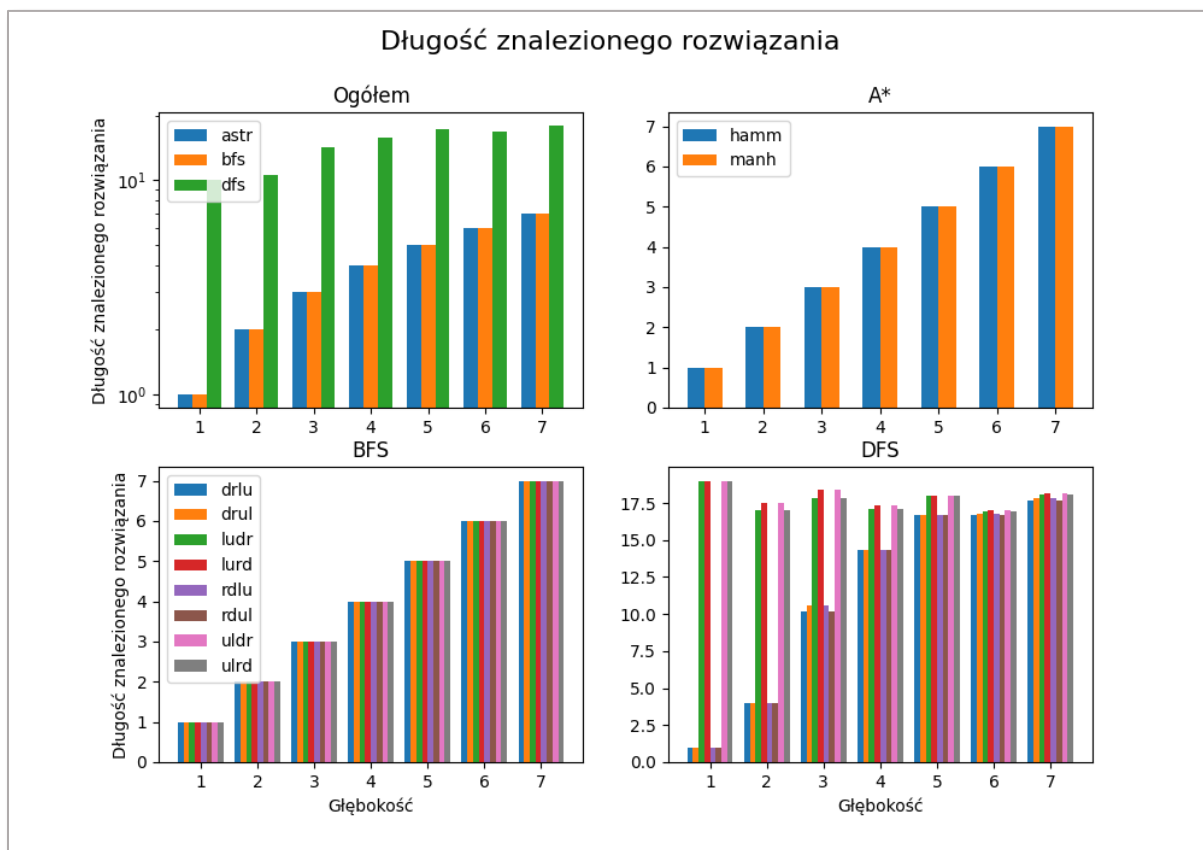
## Zadanie: Piętnastka

### 1. Cel zadania

Celem zadania było napisanie programu rozwiązującego łamigłówkę „Piętnastka” przy pomocy różnych metod przeszukiwania przestrzeni stanów, tj. strategia „wszerz”, „w głąb”. „najpierw najlepszy” – A\*. Dla ostatniej mieliśmy zastosować heurystyki: metryka Hamminga, metryka Manhattan.

Po zaimplementowaniu wszystkich niezbędnych metod przystąpiliśmy do testowania naszych rozwiązań. Po zakończonych powodzeniem testach przyszła kolej na część badawczą zadania. W tej części mieliśmy zmierzyć efektywność algorytmów zbierając dane takie jak: długość znalezionej odpowiedzi, czas potrzebny na przetworzenie łamigłówki, ilość stanów odwiedzonych i ilość stanów przetworzonych. Zebraliśmy te informacje dla wszystkich algorytmów dla plansz o odległości od 1 do 7 ruchów (do poprawnie ułożonej planszy). Dane te przedstawiliśmy w postaci wykresów słupkowych poniżej.

### 2. Wyniki



The figure consists of four bar charts arranged in a 2x2 grid, showing the number of visited states for different search algorithms across depths 1 to 7.

**Top-left chart: Ogółem (Overall)**  
 Y-axis: Liczba stanów odwiedzonych (logarithmic scale, 10<sup>0</sup> to 10<sup>6</sup>).  
 X-axis: Głębokość (Depth, 1 to 7).  
 Legend: astr (blue), bfs (orange), dfs (green).  
 Data summary: dfs is the most expensive algorithm, reaching 10<sup>6</sup> states at depth 6. bfs reaches 10<sup>2</sup> states at depth 7. astr is the least expensive, reaching 10<sup>1</sup> states at depth 7.

**Top-right chart: A\***  
 Y-axis: Liczba stanów odwiedzonych (linear scale, 0 to 25).  
 X-axis: Głębokość (Depth, 1 to 7).  
 Legend: hamm (blue), manh (orange).  
 Data summary: manh is slightly more expensive than hamm, reaching 26 states at depth 7. hamm reaches 24 states at depth 7.

**Bottom-left chart: BFS**  
 Y-axis: Liczba stanów odwiedzonych (linear scale, 0 to 400).  
 X-axis: Głębokość (Depth, 1 to 7).  
 Legend: drlu (blue), drul (orange), ludr (green), lurd (red), rdlu (purple), rdul (brown), uldr (pink), ulrd (grey).  
 Data summary: All algorithms show exponential growth in visited states. At depth 7, all algorithms reach approximately 400-450 states.

**Bottom-right chart: DFS**  
 Y-axis: Liczba stanów odwiedzonych (linear scale, 0.0 to 1.4, with a 1e6 multiplier).  
 X-axis: Głębokość (Depth, 1 to 7).  
 Legend: drlu (blue), drul (orange), ludr (green), lurd (red), rdlu (purple), rdul (brown), uldr (pink), ulrd (grey).  
 Data summary: DFS algorithms show significant variation in performance. uldr and ulrd are the most expensive, reaching 1.4e6 states at depth 6. drlu and drul are the least expensive, reaching 1.0e6 states at depth 7.

The figure consists of four bar charts arranged in a 2x2 grid, showing the number of processed states for different algorithms and depths.

**Top-left chart: Ogółem (Total)**

This chart shows the total number of processed states for three algorithms: astr (blue), bfs (orange), and dfs (green). The y-axis is logarithmic, ranging from  $10^1$  to  $10^6$ . The x-axis represents the depth (1 to 7). dfs processes the most states, followed by bfs, and then astr.

Głębokość	astr	bfs	dfs
1	~2	~2	~200,000
2	~4	~5	~400,000
3	~5	~10	~400,000
4	~7	~20	~500,000
5	~8	~40	~400,000
6	~10	~100	~600,000
7	~12	~200	~500,000

**Top-right chart: A\***

This chart shows the number of processed states for two algorithms: hamm (blue) and manh (orange). The y-axis is linear, ranging from 0 to 12. The x-axis represents the depth (1 to 7). Both algorithms show an increasing trend in the number of processed states as depth increases.

Głębokość	hamm	manh
1	2	2
2	3	3
3	4.2	4.1
4	5.1	5.2
5	6.4	6.8
6	8.0	8.8
7	10.5	12.0

**Bottom-left chart: Głębokość vs. Algorithms**

This chart shows the number of processed states for eight algorithms: drlu (blue), drul (orange), ludr (green), lurd (red), rdlu (purple), rdul (brown), uldr (pink), and ulrd (grey). The y-axis is linear, ranging from 0 to 200. The x-axis represents the depth (1 to 7). The number of processed states increases significantly with depth, with all algorithms reaching values between 190 and 210 at depth 7.

Głębokość	drlu	drul	ludr	lurd	rdlu	rdul	uldr	ulrd
1	~1	~1	~1	~1	~1	~1	~1	~1
2	~2	~2	~2	~2	~2	~2	~2	~2
3	~5	~5	~5	~5	~5	~5	~5	~5
4	~15	~15	~15	~15	~15	~15	~15	~15
5	~40	~40	~40	~40	~40	~40	~40	~40
6	~90	~90	~90	~90	~90	~90	~90	~90
7	~205	~205	~205	~205	~205	~205	~205	~205

**Bottom-right chart: A\* (hamm, manh)**

This chart shows the number of processed states for two algorithms: hamm (blue) and manh (orange). The y-axis is linear, ranging from 0 to 800,000. The x-axis represents the depth (1 to 7). Both algorithms show an increasing trend in the number of processed states as depth increases.

Głębokość	hamm	manh
1	~450,000	~450,000
2	~750,000	~750,000
3	~650,000	~650,000
4	~550,000	~550,000
5	~450,000	~450,000
6	~750,000	~750,000
7	~550,000	~550,000

[illegible]

**Czas trwania procesu obliczeniowego [ms]**

**Ogółem**

Y-axis: Czas trwania procesu obliczeniowego [ms] (log scale, 10<sup>1</sup> to 10<sup>2</sup>)

X-axis: Głębokość (1 to 7)

Legend: astr (blue), bfs (orange), dfs (green)

**A\***

Y-axis: Czas trwania procesu obliczeniowego [ms] (linear scale, 0.0 to 4.0)

X-axis: Głębokość (1 to 7)

Legend: hamh (blue), manh (orange)

**BFS**

Y-axis: Czas trwania procesu obliczeniowego [ms] (linear scale, 0.0 to 3.0)

X-axis: Głębokość (1 to 7)

Legend: drlu (blue), drul (orange), ludr (green), lurd (red), rdlu (purple), rdul (brown), uldr (pink), ulrd (grey)

**DFS**

Y-axis: Czas trwania procesu obliczeniowego [ms] (linear scale, 0 to 400)

X-axis: Głębokość (1 to 7)

Legend: drlu (blue), drul (orange), ludr (green), lurd (red), rdlu (purple), rdul (brown), uldr (pink), ulrd (grey)

### 3. Wnioski

1. Średni czas trwania procesu obliczeniowego dla algorytmu depth-first search jest znacznie wyższy niż w przypadku breadth-first search i A-star. Jest to jeden z kilku powodów, dla których możemy algorytm dfs uznać za najgorszy spośród wybranych do rozwiązywania „Piętnastki”.
2. Długość znalezionej odpowiedzi jest optymalna dla algorytmów bfs i A\*, natomiast dla dfs jest średnio dużo większa.
3. Liczba stanów odwiedzonych i przetworzonych jest najmniejsza dla A\*, a zdecydowanie największa ponownie dla dfs.
4. Zastosowane metryki nieznacznie różnią się wynikami, aczkolwiek jak już różniące występuje, to przeważnie na korzyść metryki Hamminga.