



ALGORYTMY I STRUKTURY DANYCH

STRUKTURY DANYCH: GRAFY, ALGORYTM: BFS

ALGORYTM PRZESZUKIWANIA GRAFU WSZERZ (BREADTH-FIRST SEARCH, BFS)

Nazwa algorytmu bierze się stąd, że wierzchołki w odległości k od źródła są odwiedzane przed wierzchołkami w odległości $k + 1$.

W funkcji BFS zakłada się, że graf wejściowy $G = (V, E)$ jest reprezentowany przez listy sąsiedztwa. Obliczana przez algorytm odległość od źródła do wierzchołka jest przechowywana w $dist[u]$. Poprzednik u jest pamiętany w $p[u]$. Jeśli u nie ma poprzednika, to $p[u] = None$. W algorytmie jest używana kolejka Q typu **FIFO**.

Wejście: $G = (V, E)$ – graf, s – wierzchołek początkowy (źródło).

Wyjście: drzewo przeszukiwania wszerz o korzeniu w s , dla każdego wierzchołka u osiągalnego z s obliczona odległość (najmniejsza liczba krawędzi) z s do u .

Zadanie 1. Podaj wartości d i p otrzymane w wyniku działania algorytmu przeszukiwania wszerz dla grafu skierowanego z arkusza *zadanie_1*, z wierzchołkiem 3 jako źródłem. ([Pliki do wykorzystania: analiza BFS.xlsx, arkusz zadanie_1](#)).

Zadanie 2. Podaj wartości d i p otrzymane w wyniku działania algorytmu przeszukiwania wszerz dla grafu skierowanego z arkusza *zadanie_2*, z wierzchołkiem 3 jako źródłem. ([Pliki do wykorzystania: analiza BFS.xlsx, arkusz zadanie_2](#)).

ZŁOŻONOŚĆ OBLICZENIOWA ALGORYTMU BFS.

1. Inicjowanie (w. 1 – 4) zabiera czas $O(V)$.
2. Test z w. 11 gwarantuje, że każdy wierzchołek jest wstawiany do kolejki co najwyżej raz i co najwyżej raz jest z niej usuwany. Operacje wstawiania i usuwania z kolejki zajmują czas $O(1)$. Stąd łączny czas wykonywania operacji na kolejce wynosi $O(V)$.
3. Przeglądanie wierzchołków sąsiadujących z danym wierzchołkiem (w. 10):
 - w przypadku *reprezentacji listowej* grafu, przeglądanie jest wykonywane w czasie liniowym $O(E)$
 - w przypadku *reprezentacji macierzowej* grafu, macierz sąsiedztwa każdego wierzchołka jest przeglądana co najwyżej raz. Łączny czas spędzany na przeglądaniu macierzy sąsiedztwa wynosi $O(V^2)$.

Łączny czas działania algorytmu przeszukiwania wszerz wynosi:

- ze względu na rozmiar *reprezentacji macierzowej* grafu

$$O(V + V + V^2) = O(2V + V^2) = O(V^2)$$

- ze względu na rozmiar *reprezentacji listowej* grafu

$$O(V + V + E) = O(2V + E) = O(V + E)$$



```

1  BFS(G, start)
2  for każdy wierzchołek u ∈ G
3      visited[u] = False
4      dist[u] = -1
5      p[u] = None
6
7  visited[start] = True
8  dist[start] = 0
9  Q.put(start)
10
11 while Q ≠ ∅
12     u = Q.get()
13     for v in G[u]:
14         if not visited[v]
15             visited[v] = True
16             dist[v] = dist[u] + 1
17             p[v] = u
18             Q.put(v)

```

