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 macierz sąsiedztwa. Obliczana przez algorytm odległość od źródła do wierzchołka jest przechowywana w $d[u]$. Poprzednik u jest pamiętany w $p[u]$. Jeśli u nie ma poprzednika, to $p[u]=-1$. 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 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 przeszukiwania wszerz dla grafu skierowanego z arkusza *zadanie_2*, z wierzchołkiem ? jako źródłem. (*Pliki do wykorzystania*: *analiza BFS.xlsx*, arkusz *zadanie_2*).

Złożoność obliczeniowa BFS.

Test z wiersza 12 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 $\mathrm{O}(1)$. Stąd łączny czas wykonywania operacji na kolejce wynosi $\mathrm{O}(V)$.

Macierz sąsiedztwa każdego wierzchołka jest przeglądana co najwyżej raz (w którym momencie?). Łączny czas spędzany na przeglądaniu macierzy sąsiedztwa wynosi $\mathrm{O}(V^2)$.

Inicjowanie zabiera czas $\mathrm{O}(V)$ i dlatego łączny czas działania procedury BFS wynosi

$$\mathrm{O}(V+V+V^2)=\mathrm{O}(2V+V^2)=\mathrm{O}(V^2).$$

Przeszukiwanie wszerz jest wykonywane w czasie kwadratowym ze względu na rozmiar reprezentacji macierzowej grafu. W przypadku reprezentacji listowej grafu, algorytm jest wykonywane w czasie liniowym $\mathrm{O}(V+E)$.

Bibliografia

T. H. Cormen, Ch. E. Leiserson, R. L. Rivest: *Wprowadzenie do algorytmów*. WNT Warszawa 2012.

```
BFS(G, s)
for  każdy wierzchołek u ∈ G
        visited[u] = false
        d[u] = -1
        p[u] = -1
visited[s] = true
d[s] = 0
Q.push(s)
while Q ≠ ∅
        u = Q.top()
        Q.pop()
        for v = 0 to n − 1
            if G[u][v] == 1 && !visited[v]
                visited[v] = true;
                d[v] = d[u] + 1
                p[v] = u
                Q.push(v)
```