

УДК 519.178

doi 10.54708/22259309_2026_13513

АЛГОРИТМ ФРОНТА ВОЛНЫ ДЛЯ ОРИЕНТИРОВАННОГО ГРАФА

Т. В. БАЙНАЗАРОВ¹

¹baynazarov.timurka@mail.ru

¹ФГБОУ ВО «Уфимский университет науки и технологий» (УУНИТ)

Аннотация. Алгоритм фронта волны, также известный как «поиск в ширину» (BFS, Breadth-First Search), является одним из ключевых алгоритмов теории графов. Он используется для обхода вершин графа в порядке их удалённости от стартовой вершины, что позволяет находить кратчайшие пути в невзвешенных графах. В статье будет рассмотрен алгоритм фронта волны для ориентированного графа. Также будет представлена программа, реализованная на C++, которая вычисляет расстояние, диаметр, радиус и центры в ориентированном графе.

Ключевые слова: графы; ориентированный граф; расстояния в графе; алгоритм фронт волны; центры графа; поиск в ширину.

ВВЕДЕНИЕ

Алгоритм фронта волны, также известный как поиск в ширину (BFS), является одним из фундаментальных алгоритмов теории графов. Он используется для обхода вершин графа в порядке их удалённости от стартовой вершины, что особенно полезно для нахождения кратчайших путей в невзвешенных графах. В данной статье рассматривается применение этого алгоритма для анализа ориентированных графов, включая вычисление расстояний, диаметра, радиуса и центров графа.

АЛГОРИТМ ФРОНТА ВОЛНЫ

BFS работает по принципу «первым пришёл – первым вышел», используя очередь. Алгоритм посещает все вершины, достижимые из стартовой, начиная с ближайших и постепенно удаляясь.

Шаги алгоритма:

1. Инициализация. Помещаем стартовую вершину в очередь. Задаём расстояние до неё 0, а до остальных – ∞ (или INT_MAX в C++).
2. Обход графа. Извлекаем вершину из очереди. Для всех её соседей, если они ещё не посещены, обновляем расстояние и добавляем их в очередь.
3. Завершение. Алгоритм завершается, когда очередь становится пустой.

Пример работы

Рассмотрим ориентированный граф с матрицей смежности (рис. 1):

0	1	0	0	0	1	0
1	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	1
1	1	1	1	0	0	0
0	0	0	0	1	0	1
0	0	0	1	1	0	0

Рис. 1. Матрица смежности

Стартовая вершина: 1. Порядок обхода по вершинам $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$.
 Расстояния для первой вершины: 0 1 2 3 2 1 2.

Программа, представленная в предыдущем ответе, выполняет следующие действия:

1. Читает матрицу смежности ориентированного графа.
2. Для каждой вершины запускает алгоритм фронта волны, чтобы найти расстояния до всех остальных вершин.
3. Вычисляет эксцентриситет (максимальное расстояние от вершины до остальных).
4. Находит диаметр (максимальный эксцентриситет), радиус (минимальный эксцентриситет) и центры графа (вершины с эксцентриситетом = радиусу).

Ключевые части кода:

1. Функция нахождения расстояний матрицы смежности (рис. 2).

```
vector<int> end(vector<vector<int>> &matrix, int rows, int start) {
    int temp = 0;
    int k = 0;
    vector<int> dist(rows, INT_MAX);
    queue<int> q;
    dist[start] = 0;
    q.push(start);
    while (!q.empty()) {
        int current = q.front();
        q.pop();

        for (int neighbor = 0; neighbor < rows; ++neighbor) {
            if (matrix[current][neighbor] == 1 && dist[neighbor] == INT_MAX) {
                dist[neighbor] = dist[current] + 1;
                q.push(neighbor);
            }
        }
    }
    return dist;
}
```

Рис. 2. Функция для расстояний

2. Вычисление эксцентриситета (рис. 3).

```
int ex(vector<vector<int>>& dist, int temp) {
    int maxDist = 0;
    for (int i : dist[temp]) {
        if (i != INT_MAX && i > maxDist) {
            maxDist = i;
        }
    }
    return maxDist;
}
```

Рис. 3. Функция для эксцентриситета

3. Вычисление диаметра, радиуса и центров в ориентированном графе (рис. 4).

```

cout << "\nДиаметр: ";
int maxEx = 0;
for (int i = 0; i < rows; i++) {
    if (excentr[i] > maxEx)
        maxEx = excentr[i];
}
cout << maxEx << endl;
cout << "\nРадиус: ";
int minEx = 1000;
vector<int> centre;
for (int i = 0; i < rows; i++) {
    if (excentr[i] < minEx) {
        minEx = excentr[i];
    }
}
cout << minEx << endl;
cout << "\nЦентры:\n";
for (int i = 0; i < rows; i++) {
    if (excentr[i] == minEx) {
        cout << "Вершина " << i + 1 << endl;
    }
}
}

```

Рис. 4.. Диаметр, радиус и центры

Пример работы программы (рис. 5).

```

Введите количество вершин графа: 7

Заполните матрицу графов:
0 1 0 0 0 1 0
1 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 0 0 1
1 1 1 1 0 0 0
0 0 0 0 1 0 1
0 0 0 1 1 0 0

Матрица расстояний:
0 1 2 3 2 1 2
1 0 1 2 3 2 3
4 4 0 1 3 5 2
3 3 3 0 2 4 1
1 1 1 1 0 2 2
2 2 2 2 1 0 1
2 2 2 1 1 3 0

Эксцентриситеты:
Вершина 1:3
Вершина 2:3
Вершина 3:5
Вершина 4:4
Вершина 5:2
Вершина 6:2
Вершина 7:3

Диаметр: 5

Радиус: 2

Центры:
Вершина 5
Вершина 6

```

Рис. 5. Вывод в консоли

Применение алгоритма

1. Кратчайшие пути в невзвешенных графах.
2. Анализ сетевых структур (соцсети, маршрутизация).
3. Поиск компонент связности.
4. Определение центральных узлов (например, в транспортных сетях).

Алгоритм фронта волны – мощный инструмент для анализа графов. Реализация на C++ с использованием «queue» позволяет эффективно вычислять расстояния, диаметр, радиус и центры графа.

ЗАКЛЮЧЕНИЕ

Алгоритм фронта волны (поиск в ширину, BFS) является фундаментальным инструментом в теории графов, позволяющим эффективно решать задачи, связанные с обходом вершин и анализом их взаимосвязей. В данной статье рассмотрено применение BFS для ориентированных графов, включая вычисление расстояний между вершинами, определение эксцентриситета, диаметра, радиуса и центров графа.

СПИСОК ЛИТЕРАТУРЫ

1. **Кормен, Т.**, Алгоритмы: построение и анализ, 2-е издание / Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К. – Москва: Вильямс, 2022. – 1296 с.
2. **Скиена, С. С.** Алгоритмы. Руководство по разработке / Скиена С. С. – СПб: БХВ-Петербург, 2011. – 720 с.
3. **Дейкстра, Э.** Заметки по структурному программированию // Структурное программирование / У. Дал, Э. Дейкстра, К. Хоор; пер. с англ. — М.: Мир, 1972.
4. **Поречный, С. С.**, Дискретная математика / С. С. Поречный, Н. И. Житникова, Н. М. Шерыхалина, А. Р. Ураков. – Уфа: РИК УГАТУ, 2019. – 400 с.

ОБ АВТОРАХ

БАЙНАЗАРОВ Тимур Владимирович, студент ПРО ИИМРТ УУНиТ.

METADATA

Title: The wave front algorithm for a directed graph

Author: T.V. Baynazarov

Affiliation:

¹ Ufa University of Science and Technology (UUST), Russia.

Email: ¹baynazarov.timurka@mail.ru

Language: Russian.

Source: Molodezhnyj Vestnik UGATU (scientific journal of Ufa University of Science and Technology), no. 1 (35), pp. 13-16, 2026. ISSN 2225-9309 (Print).

Abstract: The wave front algorithm, also known as breadth-First Search (BFS), is one of the key algorithms in graph theory. It is used to traverse the vertices of a graph in the order of their distance from the starting vertex, which allows you to find shortest paths in unweighted graphs. The article will consider the front-foley algorithm for a directed graph. There will also be a program implemented in C++ that calculates the distance, diameter, radius, and centers in a directed graph.

Key words: graphs, directed graph, distances in a graph, wave front algorithm, centers of a graph, breadth-first search.

About authors:

BAYNAZAROV Timur Vladimirovich, student, Dept. of COMPUTATIONAL MATHEMATICS AND CYBERNETICS (UUST).