УДК 004.421

ОЦЕНКА СЛОЖНОСТИ АЛГОРИТМА ПОИСКА КРАТЧАЙШЕГО ПУТИ ДЕЙКСТРЫ

Сабиров У.И., студент гр. ИАб-221, III курс,
Хунай-Оол Ч.А., студент гр. ИСт-221, III курс,
Семенова О.С., к.т.н., доцент
ФГБОУ ВО «Кузбасский государственный технический университет
им. Т.Ф. Горбачева»
г. Кемерово

Алгоритм Дейкстры – это классический алгоритм поиска кратчайших путей в графе с неотрицательными весами ребер. Он широко используется в различных областях, в том числе в навигационных системах, в сетях передачи данных, в компьютерных играх, в робототехнических системах, при разработке цифровых процессоров на основе технологии проектирования на кристалле[1].

Алгоритм Дейкстры работает итеративно, постепенно расширяя множество вершин с известными кратчайшими путями от начальной вершины [2]. На каждом шаге алгоритм выбирает вершину с минимальным расстоянием из множества еще не обработанных вершин, обновляет расстояния до её соседей и добавляет выбранную вершину в множество обработанных. Процесс повторяется до тех пор, пока не будут обработаны все вершины графа G(V,E), где V – множество вершин, E – множество ребер.

При выборе наиболее подходящего варианта программной реализации алгоритма Дейкстры следует учесть его временную и пространственную сложность в рамках решения конкретной задачи [3]. Сложность алгоритма Дейкстры напрямую зависит от способа реализации очереди приоритетов, используемой для хранения вершин с минимальным расстоянием. Обычно рассматриваются два основных варианта – использование массива или бинарной кучи.

В случае использования массива (рисунок 1) поиск минимального расстояния в очереди приоритетов выполняется за линейное время O(n_V), где n_V – количество вершин в графе. Обновление оценок расстояний до соседних вершин требует обхода списка смежности, что занимает время O(n_{E}), где n_{E} – количество ребер. Поскольку каждый шаг имеет сложность O(n_V) + O(n_{E}), а количество шагов равно n_V, общая временная сложность алгоритма составляет O($n_V{}^2 + n_{\text{E}}$). Так как $n_{\text{E}} \leq n_V(n_V - 1)$, то сложность алгоритма составит O($n_V{}^2$).

Бинарная куча позволяет находить минимальный элемент за время O(1), обновлять оценку расстояния до рассматриваемой вершины за время O(log n_V), обновлять оценку расстояния от рассматриваемой вершины до смежной за время O(log n_V). Так как количество шагов алгоритма равно количеству вершин n_V, а количество ребер равно n_E, то общая временная сложность алгоритма с бинарной кучей составит O(n_V log n_V + n_E log n_V), что значительно лучше, чем O($n_V{}^2$), особенно для больших и разреженных графов (где n_E << $n_V{}^2$).

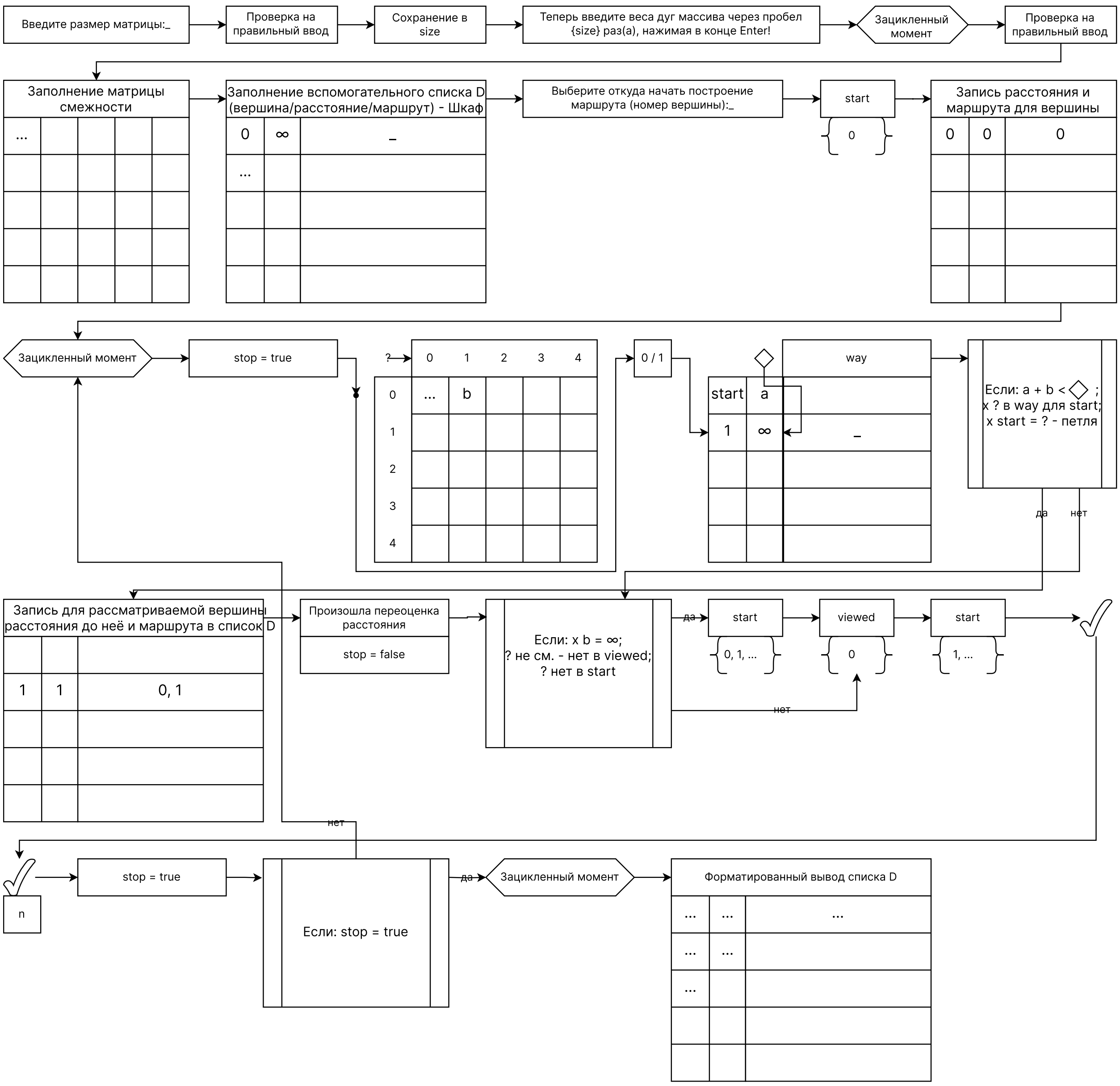

Рисунок 1 – Схема реализации алгоритма Дейкстры

Пространственная сложность алгоритма Дейкстры определяется объемом памяти, необходимым для хранения расстояний до вершин и очереди приоритетов. Для хранения расстояний требуется O(n_V) памяти. Пространственная сложность очереди приоритетов не зависит от выбранной реализации, и массив и бинарная куча занимает O(n_V) памяти. Таким образом, общая пространственная сложность алгоритма составляет O(n_V).

Сложность алгоритма Дейкстры может изменяться в зависимости от структуры графа, что подтверждается результатами экспериментов (рисунок

2, 3). Для полных графов ($n_{\mathrm{E}} \approx n_V{}^2$) использование массива приводит к сложности $\mathrm{O}(n_V{}^2)$, а бинарной кучи – к сложности $\mathrm{O}(n_V \log n_V + n_V{}^2 \log n_V)$. Для разреженных графов ($n_{\mathrm{E}} << n_V{}^2$) использование бинарной кучи обеспечивает существенное преимущество, приводя к сложности $\mathrm{O}(n_V \log n_V + n_E \log n_V)$, которая может быть значительно меньше $\mathrm{O}(n_V{}^2)$.

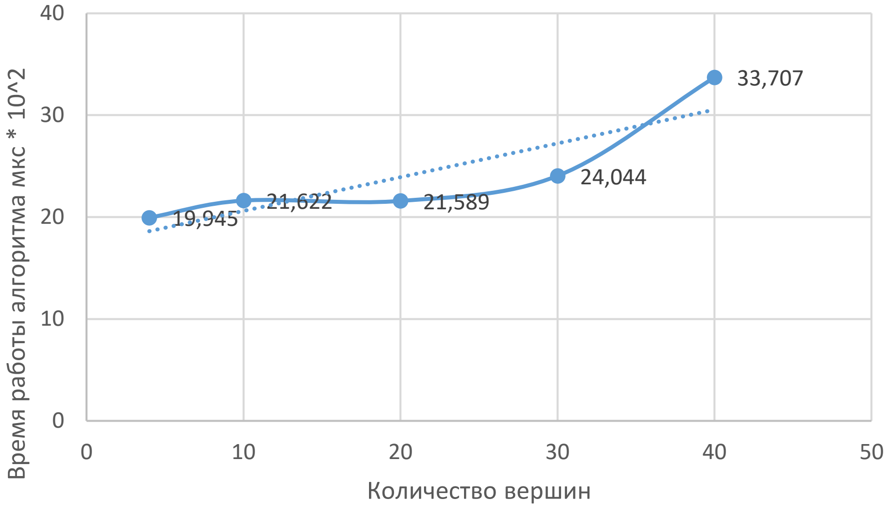

Рисунок 2 – Зависимость сложности алгоритма от количества вершин графа

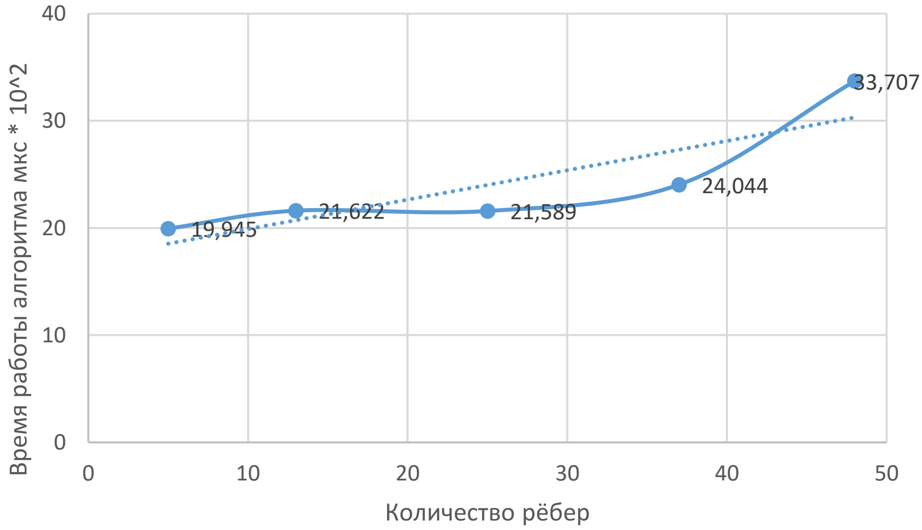

Рисунок 3 – Зависимость сложности алгоритма от количества ребер графа

Выбор между использованием массива и бинарной кучи в алгоритме Дейкстры зависит от конкретных характеристик графа. Для небольших и плотных графов использование массива может быть проще в реализации. Однако для больших и разреженных графов использование бинарной кучи значительно повышает эффективность алгоритма, приводя к асимптотически лучшей временной сложности. Понимание этих различий позволяет выбрать оптимальную реализацию алгоритма Дейкстры для конкретной задачи, обеспечивая наилучшую производительность программы.

Список литературы:

1. Грикень В. Г., Хинкель Е. Р. Оценка вычислительной сложности алгоритмов поиска пути //58 научная конференция аспирантов, магистрантов и студентов учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»: сборник тезисов докладов конференции по направлению 8: Информационные системы и технологии (Минск, 18-22 апреля 2022 года). – Минск: БГУИР, 2022.-97 с.: ил. – 2022. – С. 90.
2. Алексеев П. А. и др. Сравнительный анализ сложности алгоритмов Флойда, Дейкстры и Беллмана-Форда для графов с различным количеством вершин //Дневник науки. – 2025. – №. 1.
3. Изотова Т. Ю. Обзор алгоритмов поиска кратчайшего пути в графе //Новые информационные технологии в автоматизированных системах. – 2016. – №. 19. – С. 341-344.