ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 8

«Найти путь алгоритмами BFS, DFS, Дейкстры, A\* от собственного места постоянного проживания в СПб к университету ИТМО на переданном графе»

Выполнила работу

Автономова Ксения

Академическая группа №J3113

Принято

Ходненко Иван Владимирович

Санкт-Петербург

2024

1. Экспериментальная часть
2. Алгоритм BFS

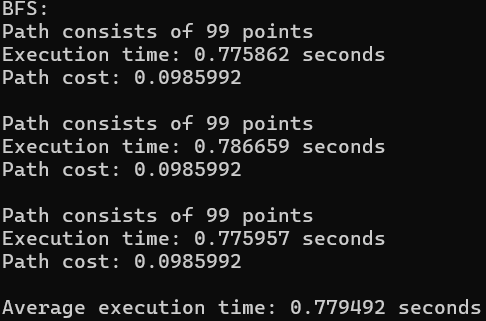
Подсчёт по памяти – примерно 16n + 32n + 17n + 16k + 16 + 17 + 16 - 16 + 16 + 16 + 17 + 16 + 16 = 65n + 16k + 114 байт.

Подсчёт асимптотики – обработка всех рёбер O(E) + O(V) = O(V + E), где V - кол-во вершин, E - кол-во рёбер.



Изображение №1 – Код алгоритма BFS.

Ниже представлены 3 запуска алгоритма BFS с выводом времени его выполнения, после чего приведено среднее общее время работы алгоритма – 0,8 секунд.



Изображение №2 - Скрин с временем работы алгоритма BFS.

1. Алгоритм DFS

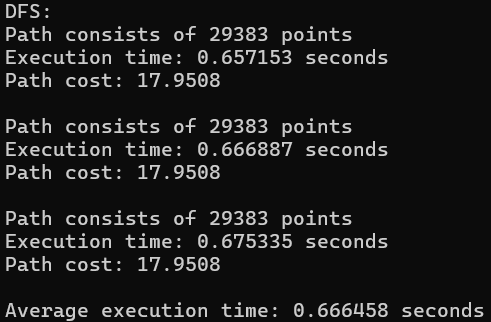
Подсчёт по памяти – примерно 16n + 32n + 16n + 16k + 16 + 16 + 16 - 16 + 16 + 16 + 16 + 16 + 16 = 64n + 16k + 112 байт.

Подсчёт асимптотики – обработка всех рёбер O(E) + O(V) = O(V + E), где V - кол-во вершин, E - кол-во рёбер.



Изображение №3 – Код алгоритма DFS.

Ниже представлены 3 запуска алгоритма DFS с выводом времени его выполнения, после чего приведено среднее общее время работы алгоритма – 0,67 секунд.

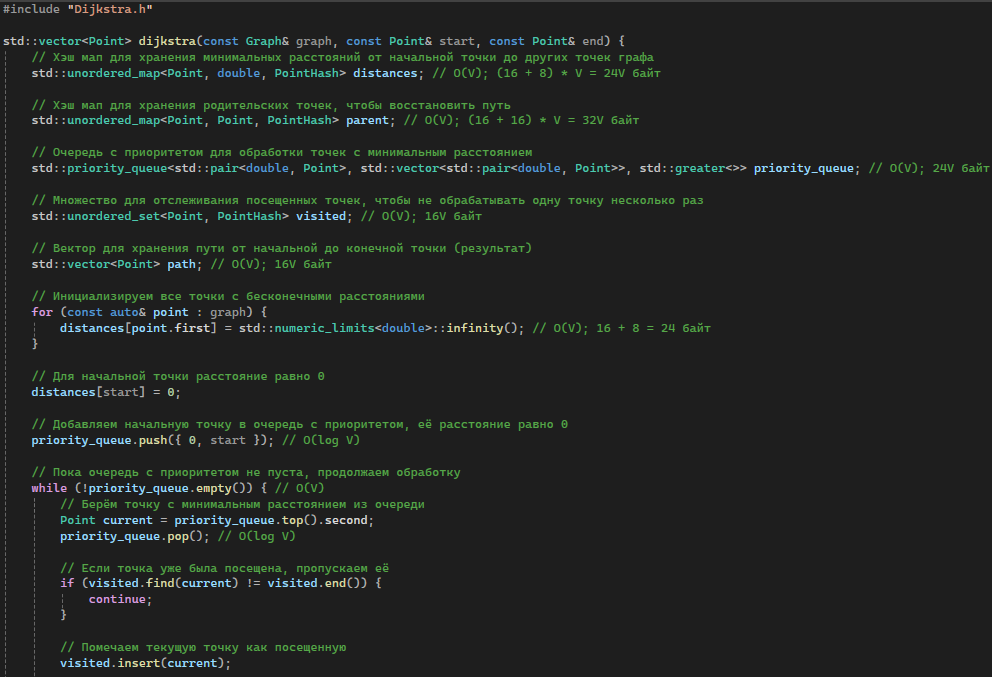


Изображение №4 - Скрин с временем работы алгоритма DFS.

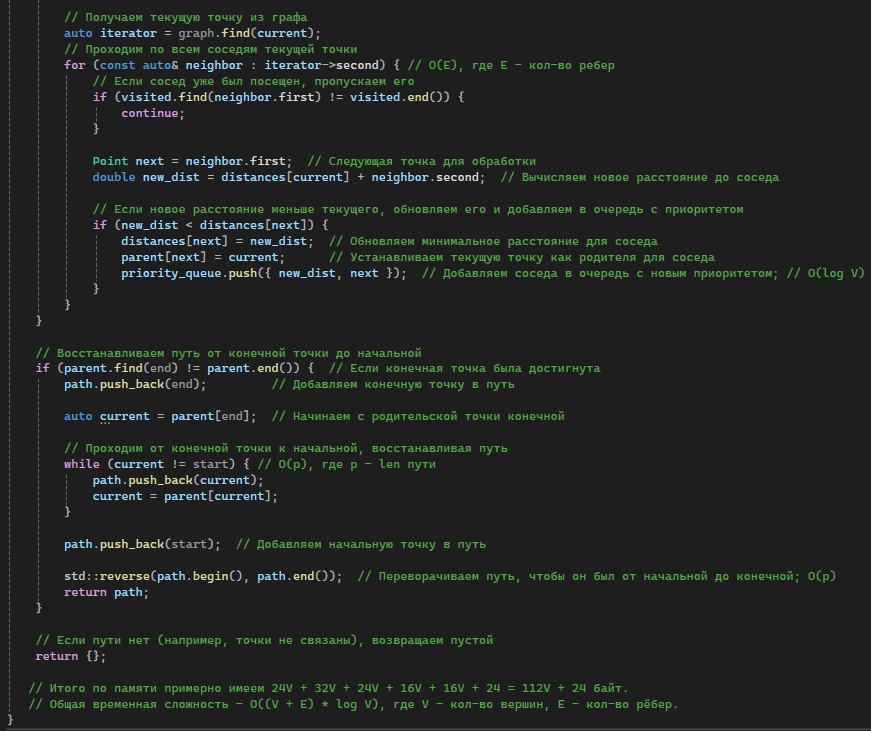
1. Алгоритм Дейкстры

Подсчёт по памяти – примерно 24V + 32V + 24V + 16V + 16V + 24 = 112V + 24 байт.

Подсчёт асимптотики – O((V + E) \* log V), где V - кол-во вершин, E - кол-во рёбер.

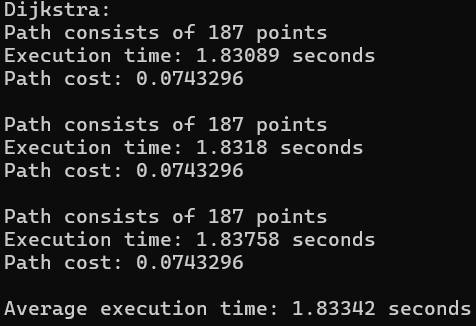


Изображение №5 – Первая часть кода алгоритма Дейкстры.



Изображение №6 – Вторая часть кода алгоритма Дейкстры.

Ниже представлены 3 запуска алгоритма Дейкстры с выводом времени его выполнения, после чего приведено среднее общее время работы алгоритма – 1,83 секунды.

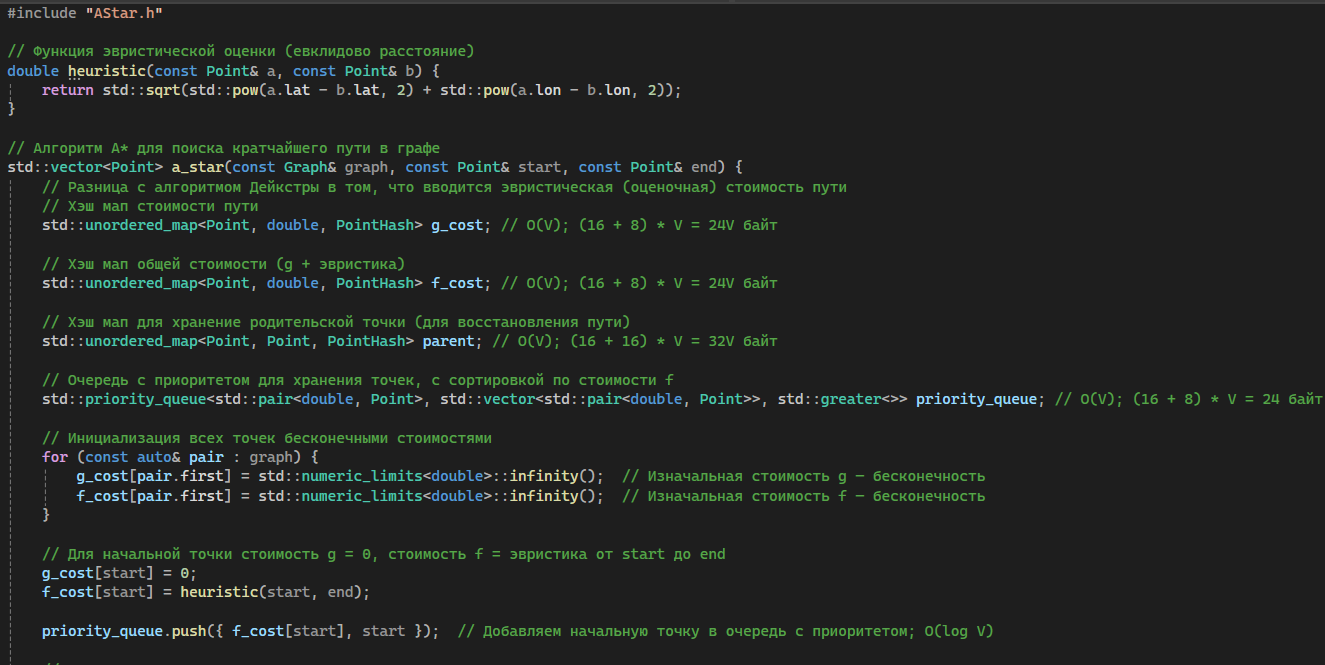


Изображение №7 - Скрин с временем работы алгоритма Дейкстры.

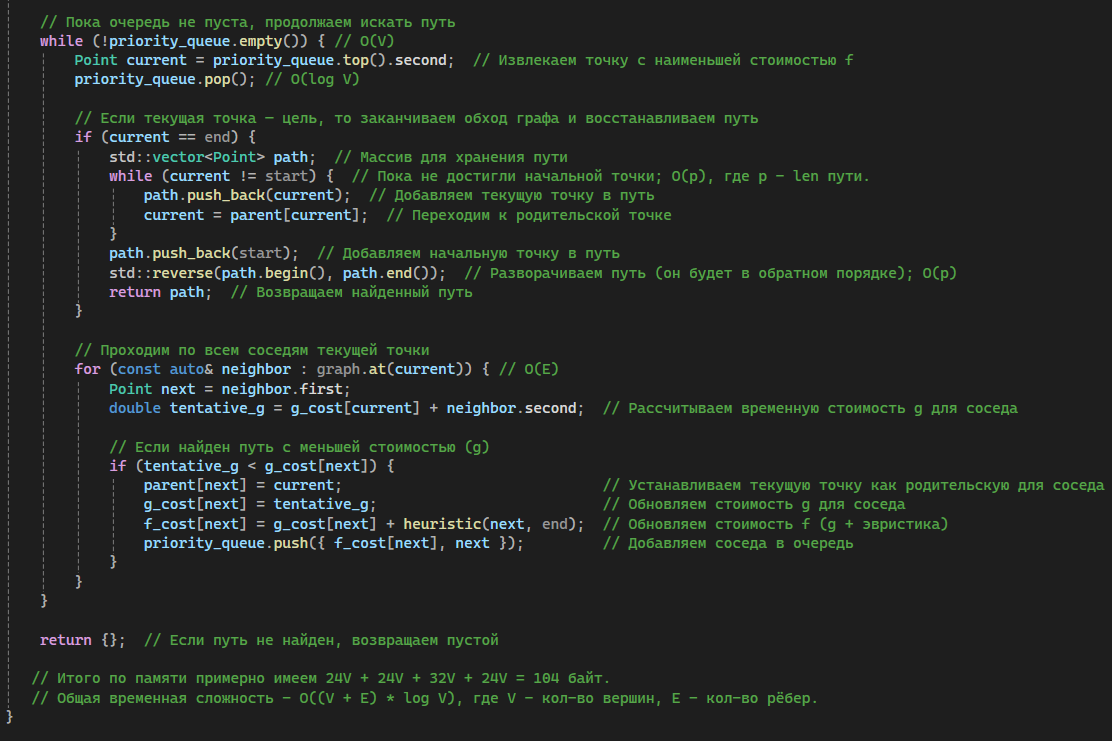
1. Алгоритм A\*

Подсчёт по памяти – примерно 24V + 24V + 32V + 24V = 104 байт.

Подсчёт асимптотики – O((V + E) \* log V), где V - кол-во вершин, E - кол-во рёбер.

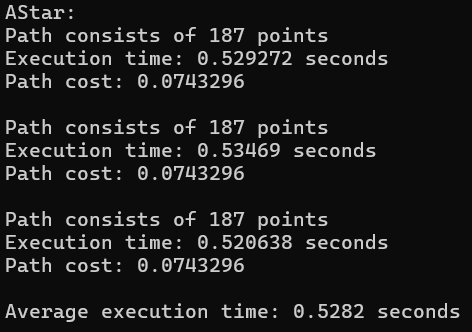


Изображение №8 – Первая часть кода алгоритма A\*.



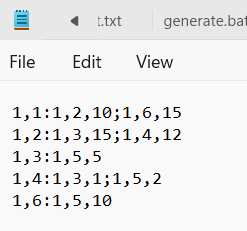
Изображение №9 – Вторая часть кода алгоритма A\*.

Ниже представлены 3 запуска алгоритма A\* с выводом времени его выполнения, после чего приведено среднее общее время работы алгоритма – 0,53 секунды.

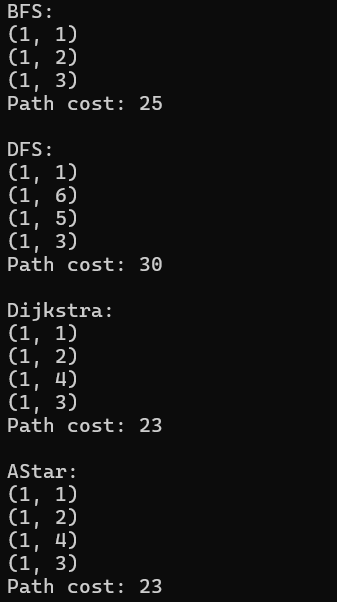


Изображение №10 - Скрин с временем работы алгоритма A\*.

1. Тесты алгоритмов: поиск маршрута из точки 1,1 в точку 1,3



Изображение №11 – Скрин тестового файла.



Изображение №12 – Скрин результатов работы алгоритмов.

Вывод: результаты работы соответствуют специфике алгоритмов.

1. Заключение

Ниже представлена таблица результатов работы алгоритмов, выполненных на файле spb\_graph.txt, для нахождения пути из точки 30.331975/59.876368 (дом) в точку 30.337795/59.926835 (ИТМО).

|  |  |  |  |
| --- | --- | --- | --- |
| Название алгоритма | Кол-во точек (вершин) | Время выполнения, сек | Вес маршрута |
| Алгоритм BFS | 99 | 0.779 | 0.098 |
| Алгоритм DFS | 29383 | 0.666 | 17.950 |
| Алгоритм Дейкстры | 187 | 1.833 | 0.074 |
| Алгоритм A\* | 187 | 0.528 | 0.074 |

Таблица №1 – Сравнение алгоритмов.

Согласно полученным результатам, алгоритм A\* сочетает в себе скорость и эффективность с точки зрения веса дуги (ребра). Остальные же алгоритмы поиска пути (BFS, DFS, Дейкстры) должны применяться, исходя из условия задачи:

- алгоритм BFS можно использовать там, где необходимо достичь точки назначения за наименьшее кол-во шагов, при этом алгоритм не учитывает вес ребер;

- алгоритм DFS подходит для решения задачи, когда требуется проверить достижимость точки назначения, но при этом эффективность маршрута не имеет значения, кроме того, память используется меньше, чем в алгоритме BFS, так как не требуется обходить все точки (вершины);

- алгоритм Дейкстры может использоваться для нахождения наиболее оптимального маршрута с учетом веса ребер, однако количество шагов при этом будет не наименьшим. С точки зрения результата работы, то он идентичен результату выполнения работы A\*, однако A\* работает значительно быстрее при правильно выбранной эвристической функции. Для задачи нахождения наиболее оптимального маршрута стоит использовать A\*, но если при этом необходимо также получить оптимальные маршруты и в другие точки, то следует использовать алгоритм Дейкстры.

1. Приложения

ПРИЛОЖЕНИЕ А

Код алгоритма BFS – BFS.cpp

Код алгоритма DFS - DFS.cpp

Код алгоритма Дейкстры - Dijkstra.cpp

Код алгоритма A\* - AStar.cpp

Дополнительные коды – LAB8.cpp, Graph.cpp