Федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

Высшая инженерная школа

**КУРСОВОЙ ПР ОЕКТ**

**Алгоритмы Дейкстры и A\***

по дисциплине

«Структуры и алгоритмы компьютерной обработки данных»

|  |  |  |
| --- | --- | --- |
| Выполнил  студент гр. ПРГ.ИС.2.2. |  | Е.В. Приходько |
| Руководитель  доцент, к.ф.-м.н. |  | В.Г. Пак |

« » 2016

Санкт-Петербург

2016

1 Постановка задачи, решаемой алгоритмами Дейкстры и A\*

Алгоритмы Дейкстры и A\* используются для поиска оптимального пути в взвешенном графе с неотрицательными весами ребер. Алгоритм Дейкстры широко применяется в протоколах маршрутизации при обменах данными по сети, для решения логистических задач и для ориентирования систем искусственного интеллекта в некотором пространстве. Алгоритм A\* похож на алгоритм Дейкстры, но использует эвристическую функцию для определения следующей рассматриваемой вершины, что позволяет уменьшить время поиска.

Рассмотри пример применения алгоритмов поиска пути. Пусть дана карта некоторой местности (рисунок 1), где серым и черным цветами обозначены, соответственно, проходимые и непроходимые клетки.

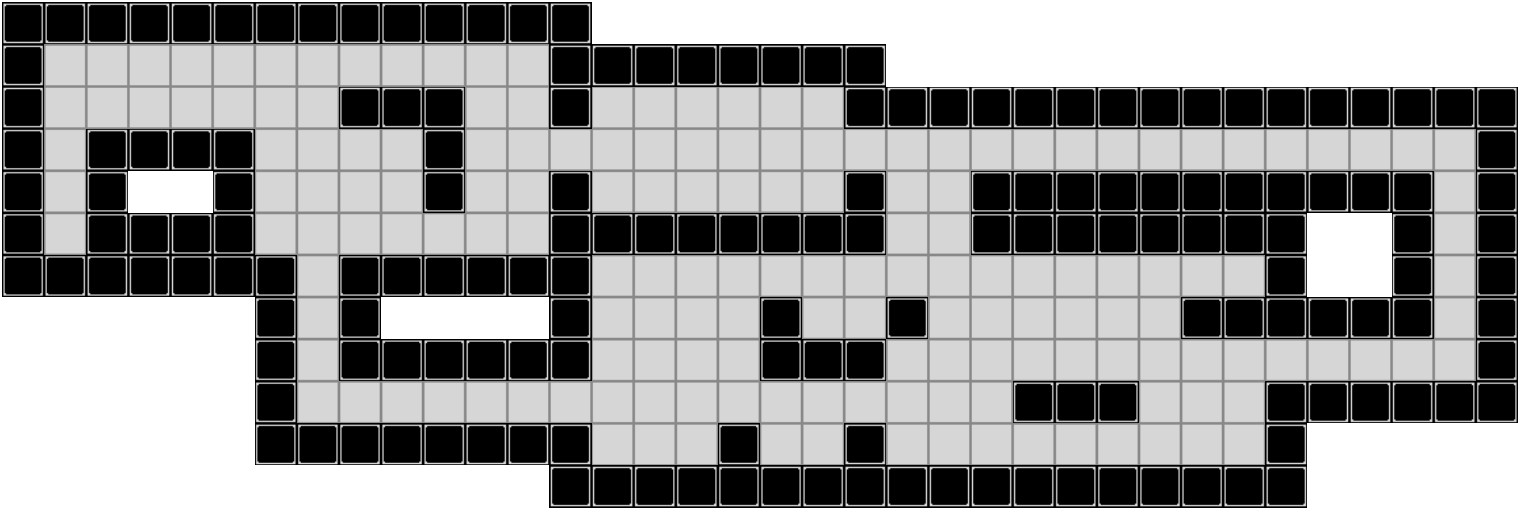


Рисунок 1. Пример топологии местности.

Сгенерировав граф возможных положений и переходов между ними, можно автоматизировать поиск пути из одной точки местности в другую (рисунок 2).

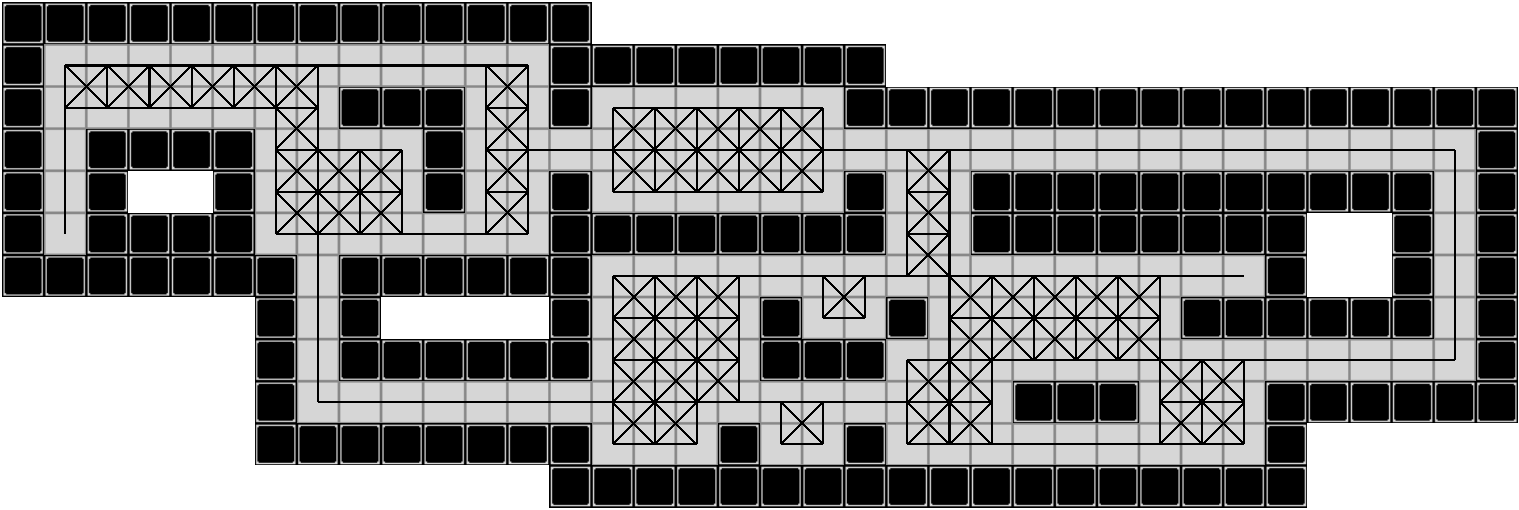


Рисунок 2. Пример местности со сгенерированным графом.

Подобный подход используется при программировании искусственного интеллекта для перемещения по реальной или виртуальной местности. В этом случае граф часто называют навигационной сеткой (navigation mesh, navmesh), пример такой сетки изображен на рисунке 3.



Рисунок 3. Пример навигационной сетки, используемой в видеоиграх.

2 Алгоритм Дейкстры

Алгоритм Дейкстры – алгоритм на графах, изобретенный нидерландским ученым Эдсгером Дейкстрой в 1959 году. В общем случае находит кратчайшие пути от выбранного узла до всех остальных.

2.1 Неформальное описание

Рассмотрим пример работы алгоритма на графе, изображенном на рисунке 4. Найдем кратчайшие пути от вершины 1 до всех остальных. Каждой вершине сопоставляется метка со значением, равным длине пути от начальной вершины до нее. Перед началом работы алгоритма метке исходной вершины устанавливается значение ноль, а всем остальным – бесконечность или другое особое значение, обозначающее отсутствие пути (рисунок 5). В последствие в ходе работы алгоритма эти значения могут уменьшаться, что означает, что был найден более короткий путь.

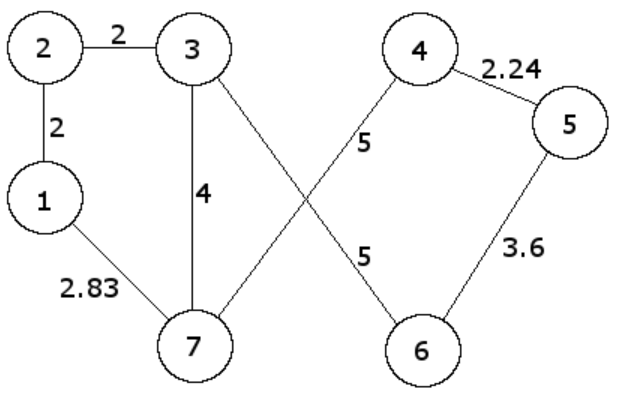


Рисунок 4. Исходный взвешенный граф.

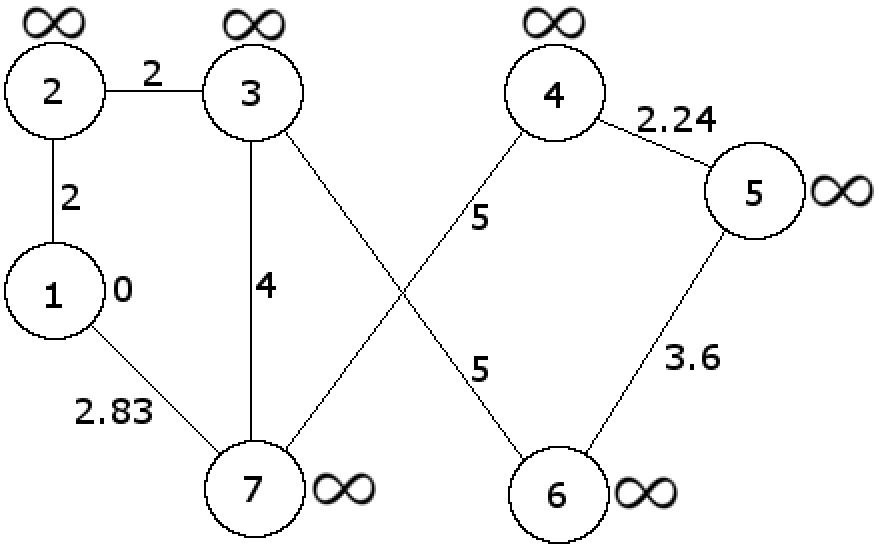


Рисунок 5. Начальное состояние меток.

На каждой итерации алгоритма вершина с минимальной меткой устанавливается в качестве текущей и помечается звездочкой, что означает, что до данной вершины найдено кратчайшее расстояние и уменьшаться она больше не будет. В данном случае выбирается вершина 1, так как ее метка равно нулю, а остальные бесконечности. Затем для всех вершин, смежных с текущей, пересчитывается значение метки: новое значение равно сумме значения текущей вершины и веса ребра между этими вершинами. Если новое значение меньше текущего, то метка обновляется. Это значит, что найден путь к этой вершине, который короче того, что был известен на данный момент.

Метка вершины 2 = min{∞, 0 + 2} = 2

Метка вершины 7 = min{∞, 0 + 2.83} = 2.83

Результат первой итерации представлен на рисунке 6.

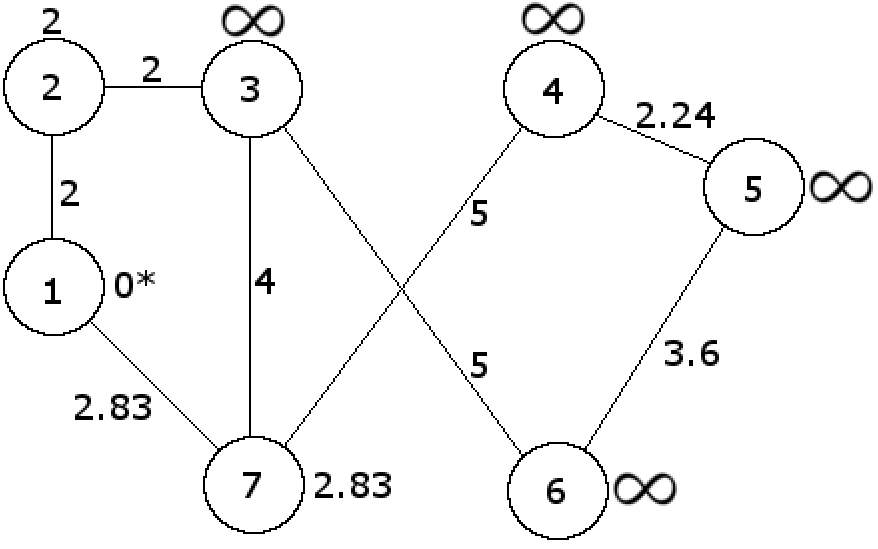


Рисунок 6. Результат первой итерации.

Выбирается новая вершина с минимальным значением метки из тех, что еще не были посещены (т.е. не отмечены звездочкой) и пересчет меток повторяется для вершин, смежных с ней. В данном примере такой вершиной является вершина 2.

Метка вершины 3 = min{∞, 2 + 2} = 4

Результат второй итерации представлен на рисунке 7.

Обход вершин продолжается, либо пока не будут посещены все вершины, либо пока не останутся вершины только со значением метки, равным бесконечности. Второй случай означает, что есть вершины, не достижимые из исходной. Результат обхода вершин из примера представлен на рисунке 8.

Для того, чтобы получить оптимальный путь из исходной вершины к другой, необходимо из целевой вершины идти в сторону начальной, каждый раз выбирая смежную вершину с минимальным значением суммы ее метки и веса ребра до текущей вершины.

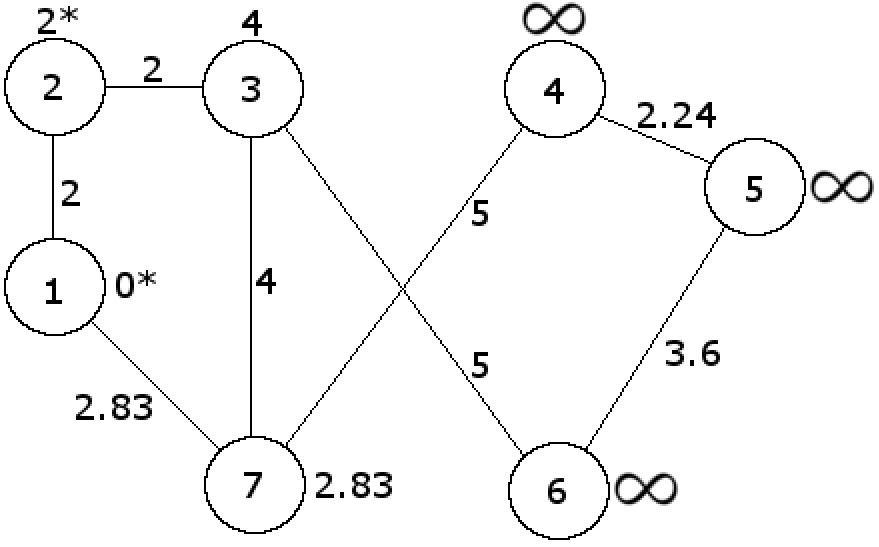


Рисунок 7. Результат второй итерации.



Рисунок 8. Итоговые значения меток.

Рассмотрим поиск оптимального пути от вершины 1 до вершины 5. Начинаем с вершины 5, перебираем все вершины, из которых можно прийти в текущую:

вершина 4: 7.83 + 2.24 = 10.07;

вершина 6: 9 + 3.6 = 12.6.

Минимальное значение у вершины 4, переходим в нее, перебираем смежные:

вершина 7: 2.83 + 5 = 5.83.

Минимальное значение у вершины 7, переходим в нее, перебираем смежные:

вершина 1: 0 + 2.83 = 2.83;

вершина 3: 4 + 4 = 8.

Минимальное значение у вершины 1, переходим в нее. Вершина 1 является исходной, значит путь найден. Таким образом, оптимальный путь из вершины 1 в вершину 5: 1-7-4-5.