- The simplest of the root refinement methods is the bisection method. Secant method uses proportional division of the interval. [3][4][5]
- In Newton's method interval of the location of the root is not defined by the initial, and its initial value. [3][4][5]
- Secant method and Newton's method have a common drawback: the accuracy of the value is checked at each step; [3][4][5]
- I came to the conclusion that Microsoft Excel spreadsheet formulas are a very powerful computing tool that allows you to quickly and easily find the root of the equation than the manual calculation;

List of sources used

1. Lectures on Numerical Analysis /Dennis Deturck, Herbert S. Wilf. — 1-е издание. — Philadelphia: Department of Mathematics University of Pennsylvania, 2002. — 125с.
2. Microsoft Excel [Электронный ресурс] : Wikipedia, The Free Encyclopedia : Version ID: 920860350, retrieved: 16 October 2019 13:27 UTC / Wikipedia contributors // Wikipedia, The Free Encyclopedia. — Электрон. дан. — Сан-Франциско: Фонд Викимедиа, 2019. — Режим доступа: https://en.wikipedia.org/w/index.php?title=Microsoft_Excel&oldid=920860350;
3. Numerical analysis [Электронный ресурс] / Wikipedia contributors. — Электрон. текстовые дан. — San Francisco: Wikipedia, The Free Encyclopedia, 2019. — Режим доступа: https://en.wikipedia.org/w/index.php?title=Numerical_analysis&oldid=895278527, свободный. — Online encyclopedia (Дата обращения: 13.05.2019);
4. Numerical methods /John D. Fenton. — 1-е издание. — Vienna: Institute of Hydraulic Engineering and Water Resources Management. Vienna University of Technology, 2019. — 33с.;
5. Numerical Methods for Physicists [Электронный ресурс] / Anthony O'Hare. — Электрон. текстовые дан. — Belton: MMHB. Department of Computer Science and Engineering, 2005. — Режим доступа: http://mars.umhb.edu/~wgt/engr2311/NMfP.pdf, свободный (Дата обращения: 12.05.2019);
6. Transcendental equation [Электронный ресурс] / Wikipedia contributors. — Электрон. текстовые дан. — San Francisco: Wikipedia, The Free Encyclopedia, 2019. — Режим доступа: https://en.wikipedia.org/w/index.php?title=Transcendental_equation&oldid=902222843, свободный. — Online encyclopedia (Дата обращения: 05.10.2019).

20.15.05

НАХОЖДЕНИЕ ВСЕХ ДОСТИЖИМЫХ ТОЧЕК (ИЗ ЗАДАННОЙ ТОЧКИ) В ПРОИЗВОЛЬНОМ ГРАФЕ

Авторы: Хрупов Денис Дмитриевич, Худяков Даниил Дмитриевич, студенты 3-го курса государственного университета «Дубна», филиал «Протвино»

Научный руководитель: Кульман Татьяна Николаевна, к.т.н., доцент кафедры Информационных технологий университета «Дубна», филиал «Протвино»

Аннотация

Исследуется произвольный граф с целью нахождения всех достижимых точек (из заданной точки) в произвольном графе.

Annotetion

An arbitrary graph is investigated in order to determine all interconnected points from a given point,

Ключевые слова: Visual Studio 2015, STL, C++ queue, arbitrary graphs, breadth-first search.

Keywords: Visual Studio 2015, STL, C++ queue.

Графы встречаются во множестве разных задач, и алгоритмы обработки графов очень важны. Существует множество разработанных алгоритмов для решения задач из разных областей человеческой деятельности. Формулировка задачи описывает, каким требованиям должно удовлетворять решение задачи, а алгоритм, решающий эту задачу, находит объект, этим требованиям удовлетворяющий. В данной работе описывается решение задачи «Нахождение всех допустимых точек в произвольном графе».

Граф – математический абстрактный объект, содержащий в себе набор вершин и ребер. Ребра представляют связи (соединения) между некоторыми парами вершин. Графы подразделяется на ориентированные, где можно двигаться по ребрам в строго заданном направлении, и неориентированные, в которых ни одному ребру не присвоено направление. Произвольный граф содержит как ориентированные ребра, имеющие одно направление, так и неориентированные ребра одновременно. Описывается граф через множества вершин и ребер

Для решения поставленной задачи анализировались разные алгоритмы.

В результате был выбран алгоритм поиска в ширину. Программа написана на языке C++ с использованием библиотеки STL, в частности, из STL взят последовательный контейнер queue (очередь).

Цель работы: исследование произвольного графа с целью нахождения всех достижимых точек (из заданной точки) в произвольном графе.

Актуальность проблемы: графы и нахождение достижимых точек в них используются во многих прикладных задачах. Например, в электрических сетях: если одна подстанция в электрической сети выходит из строя, то все связанные с ней (связный граф) так же приходят в нерабочее состояние. Если же граф несвязный, то вершины подграфов, не связанных с вышедшим из строя подграфом, остаются в работоспособном состоянии. Поэтому важно выяснить, какие узлы составляют связные подграфы в графе, чтобы иметь представление о их работоспособности.

Постановка задачи: В произвольном графе найти все точки, достижимые из заданной. Поскольку граф произвольный, то в нём выделяются несколько подграфов и связи между ними отсутствуют. При написании программы ставилась задача оптимизировать её работу.

Библиотека шаблонов (STL), – это мощный инструмент, содержащий реализацию многих структур и алгоритмов, повышающего эффективность труда программистов. При использовании STL повышается надёжность программ, их переносимость, а также уменьшаются расходы на их создание. Для решения поставленной задачи использовался поиск в ширину, реализованный с помощью очереди (queue) из библиотеки STL.

Поиск в ширину или BFS («breadth-first search» – поиск «сначала в ширину») – это обход графа, по которому после начальной вершины сначала отмечаются все вершины смежные с ней, то есть все вершины, к которым можно попасть за один шаг, а потом все те, в которые можно попасть за два шага (смежные с предыдущими и не смежные с начальной), после – за три шага и так далее.

Обход по уровням (поиск в ширину) – один из методов обхода графов. Этот метод лежит в основе нескольких других алгоритмов аналогичной тематики. Поиск в ширину предполагает поуровневый разбор графа: сперва посещается случайно подобранный узел, затем – все потомки данного узла, после этого посещаются потомки потомков и так далее. Вершины пробегаются в порядке возрастания их расстояния от корня. Рассмотрим все рёбра, отходящие от узла. Если очередной узел является целевым узлом, то поиск будет завершен; в

противном случае узел добавится в очередь. После того, как будут проверены все рёбра, отходящие от узла, из очереди исключается следующий узел, и процесс повторяется.

Поиск в ширину работает как на неориентированных, так и на ориентированных графах. Стоит отметить, что в неориентированном связном графе этот метод пройдет все существующие узлы, а в ориентированном это произойдет необязательно.

Обратимся к более формальному описанию алгоритма поиска в ширину. Основными объектами будут три структуры данных:

1. матрица смежности графа mas;
2. очередь queue;
3. массив посещенных вершин nodes.

Посещенные вершины заносятся в массив nodes – это предотвращает зацикливание, а очередь queue хранит задействованные узлы. Как известно, структура данных «очередь» работает по принципу «первый пришел – первый вышел». Далее подробно рассмотрим процесс обхода графа:

1. Массив nodes обнуляется – это означает, что в графе ни одна вершина ещё не была посещена;
2. Выбирается вершина x – стартовая вершина, которая помещается в очередь queue;
3. Вершина x исследуется (помечается как посещенная) и все смежные с ней вершины помещаются в конец очереди. Сама же вершина x удаляется;
4. Алгоритм заканчивает работу, если к этому этапу очередь становится пустой. Если этого не происходит, то посещается вершина, которая находится в начале очереди и помечается как посещенная, а все её потомки заносятся в конец очереди.

Последний пункт выполняется до тех пор, пока это возможно. Начиная со стартовой вершины, поиск в ширину постепенно уходит всё дальше и дальше, проходя уровень за уровнем. К концу работы алгоритма будут найдены все наикратчайшие пути из стартовой вершины до каждого узла, к которому из неё можно прийти.

Схема алгоритма поиска в ширину. Поиск в ширину выполняется в следующем порядке: началу обхода – вершине x приписывается метка «0», а вершинам, смежным с ней, – метка «1». После чего поочередно рассматривается окружение всех вершин, которым присвоена метка «1» – и к каждой вершине, которая входит в подобное окружение, приписываем метку «2» и так далее. Если исходный граф связный, то поиск в ширину даст метки всем его вершинам. Дуги вида (i, i+1) порождают остовный бесконтурный орграф, который содержит в качестве своей части остовное ордерево, называемое также поисковым деревом. Заметим, что с помощью поиска в ширину можно также занумеровать вершины, нумеруя вначале вершины с меткой «1», затем с меткой «2» и так далее.

Пример реализации. Рассмотрим задачу: в произвольном графе (рис. 1) найти все достижимые точки из заданной начальной вершины (вводится пользователем вручную с клавиатуры).

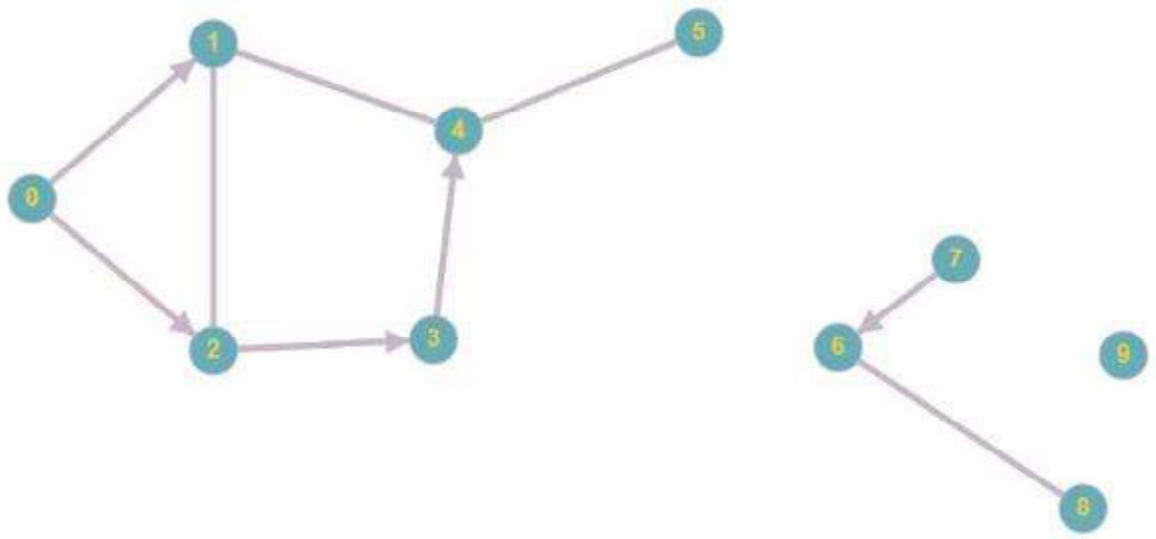

Рис. 1 – заданный граф

Матрица смежности – это вид представления графа в виде матрицы, в котором пересечение столбцов и строк задают дуги. Матрицей смежности можно задать вес дуг и ориентацию. Каждая строка и столбец матрицы соответствуют вершинам, номер строки соответствует вершине, из которой выходит дуга, а номер столбца – той вершине, в какую входит дуга (рис. 2).

```
0, 1, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0, 1, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

Рис. 2 – матрица смежности

Итак, введя в качестве первоначальной вершины узел 1, пользователь имеет следующий результат работы программы (рис. 3):

```
Введите начальную вершину:
1
----------------------------
2 4      - вершины доступные из  1
1 3      - вершины доступные из  2
1 5      - вершины доступные из  4
4        - вершины доступные из  3
4        - вершины доступные из  5
Для продолжения нажмите любую клавишу . . .
```

Рис. 3 – результат работы программы

На (рис.3) результат работы программы. Задача решена, из заданной вершины графа находятся достижимые вершины.

Заключение. В результате получен и реализован на языке C++ оптимальный алгоритм для нахождения всех достижимых точек (из заданной точки) в произвольном графе. В этой работе был исследован принцип работы алгоритма поиска в ширину.

Список использованных источников

1. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженеров. - М.: Энергоатомиздат, 1988. - 480 с.
2. Коршунов Ю.М. Математические основы кибернетики: Учеб. Пособие для вузов. - 3-е изд. перераб. и доп. - М.: Энергоатомиздат, 1987. - 496 с.: ил.
3. Новиков Ф.А. Дискретная математика для программистов. - Спб: Питер, 2000. - 304 с.: ил.
4. Яблонский С.В. Введение в дискретную математику: Учебное пособие для Вузов/ Под ред.В.А. Садовничего - 3-е изд. стер. - М.: Высш. шк., 2001. - 384 с.
5. Липский В. Комбинаторика для программистов. М.: Мир, 1988. - 213 С. .
6. Кристофидес Р. Теория графов. Алгоритмический подход. М.: Мир, 1978. - 432 с.

28.27.15

МАТЕМАТИЧЕСКОЕ ОБОСНОВАНИЕ МЕР ПО УЛУЧШЕНИЮ БЕЗОПАСНОСТИ СИСТЕМЫ

Автор: Хусаинов Владимир Алексеевич, студент 3 курс, Московский Авиационный Институт

Научный руководитель: Олейников Владимир Петрович., доцент кафедры 808Б к.ф.-м.н.

Аннотация

Рассмотрен алгоритм улучшения безопасности системы управления. Оценка недостатков применения различных методов повышения безопасности.

Annotation

Consideration of an algorithm for improving the security of the control system. Assessing the disadvantages of using various methods to improve security.

Ключевые слова: Системы управления, надёжность систем, отказ.

Keywords: Control systems, system reliability, failure.

Введение

Система управления представляет собой совокупность взаимосвязанных и взаимозависимых элементов, образующих упорядоченную целостность, единство. Основой упорядочения системы управления является, как правило, цель ее функционирования.

Существует множество методов увеличения надёжности систем управления, однако они могут быть сведены к нескольким основным.