возможные сроки «стагнации», а потери от падения производительности труда стараться сводить к минимальному уровню.

Отсутствие единого стандарта проектирования также является одной из основных проблем внедрения BIM технологий. В любой компании, переходящей на BIM, рано или поздно появляются типовые проблемы: разные подходы к работе у специалистов, использование различных параметров семейств для одних и тех же свойств, сложности с организацией схемы совместной работы и так далее. Все эти пункты снижают эффективность применения программы. Для решения этих проблем существует специалист – BIM-менеджер, который разбирается с затруднениями, и, во избежание проблем, формирует BIM-стандарт компании. В стандарте указывается кто, как и какими инструментами должен работать в Revit, какие файлы из какой библиотеки использовать, как открывать файлы для просмотра, печатать их и так далее. Нужно стремиться к тому, чтобы при любом вопросе пользователь мог обратиться к BIM-стандарту и получить ответ. BIM-стандарт является уникальным документом и разрабатывается индивидуально для каждой организации. Основой для разработки такого документа являются результаты нескольких выполненных пилотных проектов, в процессе работы, над которыми были выявлены основные аспекты и сложности в переходе к BIM.

1. Куприяновский В.П., Синягов С.А., А.П. Добрынин BIM- Цифровая экономика. Как достигли успеха? Практический подход к теоретической концепции [Электронный ресурс]. Режим доступа: <http://injoit.org/index.php/j1/article/view/275>
2. А.Е. Мамаев, В.В. Шарманов, Ю.С. Золотова, В.А. Свиницкий, Г.С. Городнюк – Прикладное применение BIM-модели здания для контроля инвестиционно-строительного проекта // Актуальные проблемы гуманитарных и естественных наук. – 2016. – № 1 (3). – С. 83-87.
3. Козлов И.М. Оценка экономической эффективности внедрения BIM [Электронный ресурс]. — Режим доступа: http://www.integralsib.ru/articles/vnedrenie_bim/economy/
4. Рыбин Е.Н., Амбарян С.К., Аносов В.В., Гальцев Д.В., Фахротов Н.А. BIM – технологии / Е.Н. Рыбин, С.К. Амбарян, В.В. Аносов, Д.В. Гальцев, Н.А. Фахротов. — BIM - технологии // Известия вузов. Инвестиции. Строительство. Недвижимость. — 2019. — № 9. — С. 98-105.
5. Лустина, О. В. Использование BIM-технологий в современном строительстве / О. В. Лустина, Н. А. Бикбаева, А. М. Купчиков. — Использование BIM - технологий в современном строительстве // Молодой ученый. — 2016. — № 15 (119). — С. 187-190.

Кравченко К.И., Минеева Т.А.

Алгоритм Дейкстры для определения кратчайшего маршрута следования

*Северный (Арктический) федеральный университет имени М. В. Ломоносова
(Россия, Архангельск)*

doi: 10.18411/lj-04-2021-12

Аннотация

В данной статье рассматривается алгоритм Дейкстры и его применение для нахождения кратчайшего маршрута. В статье описывается алгоритм Дейкстры, приводится пример программы на языке C++, позволяющей найти кратчайший маршрут.

Ключевые слова: нахождение кратчайшего маршрута, кратчайший путь, алгоритм Дейкстры, неориентированный граф, матрица достижимости.

Abstract

This article discusses Dijkstra's algorithm and its application to find the shortest route. The article describes Dijkstra's algorithm, provides an example of a C++ program that allows you to find the shortest route.

Keywords: finding the shortest route, shortest way, Dijkstra's algorithm, undirected graph, reachability matrix.

Задача нахождения кратчайшего маршрута находит широкое отражение в различных областях деятельности человека: картографические сервисы, транспортная логистика, коммуникационные сети и т. д. Данная работа имеет своей целью изучение возможности применения алгоритма Дейкстры для разработки программы, предназначенной для определения кратчайшего маршрута следования. Основными задачами нашего исследования являются: изучение теоретического описания алгоритма Дейкстры, реализация программы по нахождению кратчайшего маршрута на языке C++ и практическое применения разработанной нами программы.

Алгоритм Дейкстры – алгоритм на графах, изобретённый нидерландским учёным Эдсгером Дейкстрой в 1959 году. Находит кратчайшие пути от одной из вершин графа до всех остальных. Алгоритм работает только для графов без рёбер отрицательного веса [3].

Отличительной особенностью в реализации данного алгоритма является применение двух массивов. Один из них предназначен для хранения информации о посещенных вершинах, изначально все вершины отмечены как не посещенные. Во втором будут находиться найденные кратчайшие пути для конкретной вершины, изначально в элементы массива будет занесено недостижимо большое число. Расстояния от одной вершины до другой записываются в матрицу достижимости. Посещая каждую вершину, мы отмечаем её как посещённую и находим расстояния от этой вершины до соседних, если найденное расстояние является кратчайшим, то оно замещает имеющееся в массиве значение расстояния до этой вершины. Таким образом, мы находим кратчайшие пути от начальной вершины до всех остальных.

Для нахождения кратчайшего маршрута нам требуется доработать алгоритм Дейкстры: необходимо сверить разность расстояния до конечной вершины и расстояний от конечной до соседних с кратчайшим расстоянием, записанным в массиве для этой вершины, если в результате мы получим значение, которое совпадает с длиной пути рассматриваемой вершины, то именно из нее был осуществлен переход в конечную вершину. Таким образом, мы восстанавливаем маршрут, пока не дойдём до начальной вершины.

Рассмотрим реализацию программы для нахождения кратчайшего маршрута следования на одном из языков программирования. Мною был выбран язык C++ в среде Visual Studio от компании Microsoft.

Ниже представлен код программы, предназначенной для нахождения кратчайшего маршрута следования, основанный на принципе работы алгоритма Дейкстры.

Листинг 1

Программа по нахождению кратчайшего маршрута на языке C++

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
constexpr auto Vertices = 13;
int main()
{
    int Reachability_Matrix[Vertices][Vertices];
    int Distance[Vertices];
    int Visited[Vertices];
    int Temp, Min_Index, Min;
    int Begin_Vertex = 0;
    system("chcp 1251");
    system("cls");
    for (int i = 0; i < Vertices; i++)
    {
        Reachability_Matrix[i][i] = 0;
        for (int j = i + 1; j < Vertices; j++) {
```

```
        printf("Введите расстояние %d - %d: ", i + 1, j + 1);
        scanf("%d", &Temp);
        Reachability_Matrix[i][j] = Temp;
        Reachability_Matrix[j][i] = Temp;
    }
}
for (int i = 0; i < Vertices; i++)
{
    Distance[i] = 10000;
    Visited[i] = 1;
}
Distance[Begin_Vertex] = 0;
do {
    Min_Index = 10000;
    Min = 10000;
    for (int i = 0; i < Vertices; i++)
    {
        if ((Visited[i] == 1) && (Distance[i] < Min))
        {
            Min = Distance[i];
            Min_Index = i;
        }
    }
    if (Min_Index != 10000)
    {
        for (int i = 0; i < Vertices; i++)
        {
            if (Reachability_Matrix[Min_Index][i] > 0)
            {
                Temp = Min + Reachability_Matrix[Min_Index][i];
                if (Temp < Distance[i])
                {
                    Distance[i] = Temp;
                }
            }
        }
        Visited[Min_Index] = 0;
    }
} while (Min_Index < 10000);
int Ver[Vertices];
int End_Vertex = 12;
Ver[0] = End_Vertex + 1;
int k = 1;
int Weight = Distance[End_Vertex];
while (End_Vertex != Begin_Vertex)
{
    for (int i = 0; i < Vertices; i++)
    {
        if (Reachability_Matrix[i][End_Vertex] != 0)
        {
            int Temp = Weight - Reachability_Matrix[i][End_Vertex];
            if (Temp == Distance[i])
            {
                Weight = Temp;
                End_Vertex = i;
                Ver[k] = i + 1;
                k++;
            }
        }
    }
}
printf("\nКратчайший маршрут:\n");
for (int i = k - 1; i >= 0; i--)
```

```
printf("%3d ", Ver[i]);  
getchar(); getchar();  
return 0;  
}
```

Одним из возможных мест использования разработанной нами программы является её применение в работе картографических сервисах для нахождения кратчайшего пути между двумя точками на карте. Для этого необходимо представить карту местности, как неориентированный граф.

Рисунок 1 – Карта города Северодвинска

За вершины в данном случае будут приниматься каждое пересечение двух или более дорог, а также начальная и конечная точка. За рёбра графа будут приниматься дороги.

Рисунок 2 – Неориентированный граф

На основании полученного графа необходимо сформировать матрицу достижимости, в которой номера строк и столбцов соответствуют вершинам графа, а элементы матрицы – длинам рёбер графа.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	2	4	0	0	0	0	0	0	0	0	0	0
2	2	0	0	3	0	0	0	0	0	0	0	0	0
3	4	0	0	0	0	3	0	0	0	0	0	0	0
4	0	3	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	3	0	0	0	0	0
6	0	0	3	0	0	0	0	0	2	0	0	0	0
7	0	0	0	0	0	0	0	1	0	2	0	0	0
8	0	0	0	0	3	0	1	0	2	0	0	0	0
9	0	0	0	0	0	2	0	2	0	0	0	0	2
10	0	0	0	0	0	0	2	0	0	0	2	0	0
11	0	0	0	0	0	0	0	0	0	2	0	3	0
12	0	0	0	0	0	0	0	0	0	0	3	0	2
13	0	0	0	0	0	0	0	0	2	0	0	2	0

Рисунок 3 – Матрица достижимости

Результат работы программы по нахождению кратчайшего маршрута представлен на рисунке 4.

Кратчайший маршрут:
1 3 6 9 13

Рисунок 4 – Результат работы программы по нахождению кратчайшего маршрута

Согласно поставленной цели, был изучен материал о принципах работы алгоритма Дейкстры, разработана программа, позволяющая найти кратчайший маршрут между начальной и конечной точкой. Практическое применение разработанной нами программы на языке C++ было рассмотрено и реализовано на примере её работы в картографических сервисах.

1. Microsoft [Электронный ресурс] / Документация по C++ – режим доступа: <https://docs.microsoft.com/ru-ru/cpp/?view=msvc-160>, свободный (Дата обращения: 20.03.2021)
2. Асанов М.О., Баранский В.А., Расин В.В., Дискретная математика графы, матроиды, алгоритмы. — Санкт-Петербург: Лань, 2020. — 364 с
3. Wikipedia [Электронный ресурс] / Алгоритм Дейкстры – режим доступа: https://ru.wikipedia.org/wiki/Алгоритм_Дейкстры, свободный (Дата обращения: 20.03.2021)

Лысенко В.А., Перова М.В.

СЭД как ключевой инструмент организации удаленной работы сотрудников

*Южно-Российский институт управления – филиал РАНХиГС
(Россия, Ростов-на-Дону)*

doi: 10.18411/lj-04-2021-13

Аннотация

Данная статья посвящена определению необходимости внедрения систем электронного документооборота (СЭД) для удаленной работы сотрудников. Рассмотрены варианты и перспективы такой работы с СЭД «Дело», EOS for SharePoint и EOSmobile.

Ключевые слова. СЭД, удаленная работа, СЭД «Дело», документация, EOS for SharePoint, EOSmobile.