Учреждение Образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Лабораторная работа № 2

**«АЛГОРИТМЫ НА ГРАФАХ»**

По дисциплине

«Математическое программирование»

Выполнил:

Студент ФИТ

Специальности ИСиТ 2 курса 3 группы

Шастовская Марина Сергеевна

Вариант 10

Минск 2022

**Лабораторная работа 5**

**АЛГОРИТМЫ НА ГРАФАХ**

**(алгоритмы поиска в ширину и глубину, топологическая сортировка)**

**Цель работы:** освоить сущность и программную реализацию: а) способов представления графов; б) алгоритмов поиска в ширину и глубину; в) алгоритма топологической сортировки графов.

**Поиск** **в** **глубину** – это обход графа по возможным путям, когда нужно сначала полностью исследовать одну ветку и только потом переходить к другим веткам (если они останутся нерассмотренными). **Поиск** **в** **ширину** – это обход графа по возможным путям, когда после посещения вершины, посещаются все соседние с ней вершины.

**Стратегия поиска в глубину:**

состоит в том, чтобы идти «вглубь» графа, насколько это возможно.

 Алгоритм: перебираем все исходящие из рассматриваемой вершины рёбра. Если ребро ведёт в вершину, которая не была рассмотрена ранее, то запускаем алгоритм от этой нерассмотренной вершины, а после возвращаемся и продолжаем перебирать рёбра. Возврат происходит в том случае, если в рассматриваемой вершине не осталось рёбер, которые ведут в нерассмотренную вершину. Если после завершения алгоритма не все вершины были рассмотрены, то необходимо запустить алгоритм от одной из нерассмотренных вершин[[](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B8%D1%81%D0%BA_%D0%B2_%D0%B3%D0%BB%D1%83%D0%B1%D0%B8%D0%BD%D1%83#cite_note-_7b9873a2d90ce124-1)

**Стратегия поиска в ширину:**

В процессе обхода мы идём вширь. он посещает все вершины графа по одному разу

*Задача топологической сортировки* графа состоит в следующем: указать такой линейный порядок на его вершинах, чтобы любое ребро вело от вершины с меньшим номером к вершине с большим номером.(поиском в глубину)

**ЗАДАНИЕ ДЛЯ ВЫПОЛНЕНИЯ:**

**Задание 1.**  Ориентированный граф **G** взять в соответствии с вариантом. Осуществить алгоритмы поиска в ширину и глубину, а также алгоритма топологической сортировки аналогично примерам, рассмотренным на лекциях. Оформить отчет, включив в него **каждый** шаг выполнения алгоритмов.

**Выполнение**

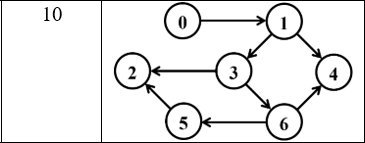


Рис.1 граф 10 вариант

Граф: (0,1), (1,4), (1,3), (3,6), (3,2), (6,4), (6,5), (5,2)

Текущее состояние алгоритма хранится в следующих структурах памяти:

Q – очередь вершин,

C – массив окраски вершин,

D – массив расстояний,

P – массив предшествующих вершин(те которые прошли)

1. **Алгоритм поиска в ширину (BFS).**

**Алгоритм поиска в ширину** (англ. breadth-first search, BFS) позволяет найти кратчайшие пути из одной вершины невзвешенного (ориентированного или неориентированного) графа до всех остальных вершин. Под кратчайшим путем подразумевается путь, содержащий наименьшее число ребер.

По условию, граф имеет 7 вершин, пронумерованных начиная с нуля. В качестве стартовой вершины выбрана вершина с номером 0.

Q – очередь вершин,

C – массив окраски вершин,

D – массив расстояний,

P – массив предшествующих вершин (те которые прошли)

Шаг 1. В качестве стартовой вершины выбираем вершину с номером 0.

0

1

2 3 4

5 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Q | 0 |  |  |  |  |  |  |
| C | B | W | W | W | W | W | W |
| D | 0 | I | I | I | I | I | I |
| P | N | N | N | N | N | N | N |

Шаг 2. У вершины 0 одна смежная вершина, значит выбираем 1. Закрашиваю нулевую в черный цвет, как пройденную. А смежную– в серый.

0

2 3 4

5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Q | 1 |  |  |  |  |  |  |
| C | B | G | W | W | W | W | W |
| D | 0 | I | I | I | I | I | I |
| P | N | 0 | N | N | N | N | N |

Шаг 3. У вершины 1 две смежные вершины; для последующего пути выбираю вершину с наименьшим весом из двух – четвёртую, а вторую добавляю в начало очереди.

0

2 3 4

5 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Q | 3 | 4 |  |  |  |  |  |
| C | B | B | G | G | W | W | W |
| D | 0 | 1 | 0 | I | I | I | I |
| P | N | 0 | N | 1 | 1 | N | N |

Шаг 4. Перехожу в смежную вершину – вершина номер три и четыре. С тройкой есть три смежные вершины – два, пять и шесть, закрашиваю их в серый цвет для попадания в очередь, а третью в черный цвет как пройденную.

0

2 3 4

5 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Q | 2 | 5 | 6 |  |  |  |  |
| C | B | B | G | B | B | G | G |
| D | 0 | 1 | 1 | 2 | 3 | 2 | 3 |
| P | N | 0 | 3 | 3 | 1 | 3 | 3 |

Шаг 5. Перехожу в смежную вершину – вершина номер шесть. Единственная смежная с ней вершина номер четыре, закрашиваю шестую и четвёртую в чёрный цвет как пройденную.

1

0

0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Q | 5 | 6 |  |  |  |  |  |
| C | B | B | B | B | B | B | B |
| D | 0 | 1 | 1 | 2 | 3 | 2 | 3 |
| P | N | 0 | 3 | 3 | 1 | 3 | 3 |

В результате получили BFS-дерево:0,1,3,4,2,5,6

**Алгоритм поиска в глубину (DFS).**

**Обход**означает посещение всех узлов графа. **«Обход в глубину» или «Поиск в глубину»**- это рекурсивный алгоритм поиска всех вершин графа или древовидной структуры данных.

***Поиск в глубину****— один из методов обхода*[*графа*](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0))*. Стратегия поиска в глубину, как и следует из названия, состоит в том, чтобы идти «вглубь» графа, насколько это возможно. Алгоритм поиска описывается рекурсивно:*

*Если ребро ведёт в вершину, которая не была рассмотрена ранее, то запускаем алгоритм от этой нерассмотренной вершины, а после возвращаемся и продолжаем перебирать рёбра.*

*Возврат происходит в том случае, если в рассматриваемой вершине не осталось рёбер, которые ведут в нерассмотренную вершину. Если после завершения алгоритма не все вершины были рассмотрены, то необходимо запустить алгоритм от одной из нерассмотренных вершин*[*[1]*](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B8%D1%81%D0%BA_%D0%B2_%D0%B3%D0%BB%D1%83%D0%B1%D0%B8%D0%BD%D1%83#cite_note-_7b9873a2d90ce124-1)*.*

По условию, граф имеет 7 вершин, пронумерованных начиная с нуля. В качестве стартовой вершины выбрана вершина с номером 0.

Текущее состояние алгоритма хранится в следующих структурах памяти:

C – массив окраски вершин,

D – время окраски вершин в серый цвет,

P – массив предшествующих вершин,

F – время окраски в чёрный цвет,

t – номер шага алгоритма.

Шаг 1

0 1

2 3 4

t = 1 – стартовый шаг

5 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **С** | G | W | W | W | W | W | W |
| **D** | 0 | I | I | I | I | I | I |
| **P** | 0 | N | N | N | N | N | N |
| **F** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Шаг 2

0 1

2 3 4

t = 2

5 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **С** | G | G | W | W | W | W | W |
| **D** | 0 | 1 | I | I | I | I | I |
| **P** | N | 1 | N | N | N | N | N |
| **F** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Шаг 3

0 1

2 3

t = 3

5 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **С** | G | G | W | G | B | W | W |
| **D** | 0 | 1 | I | I | 3 | I | I |
| **P** | 0 | 1 | N | 2 | N | N | N |
| **F** | 0 | 0 | 0 | 0 | 3 | 0 | 0 |

Шаг 4

0 1

2

t = 4

5 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **С** | G | G | G | G | B | W | W |
| **D** | 0 | 1 | I | 4 | 3 | I | I |
| **P** | N | 0 | 3 | 2 | N | N | N |
| **F** | 0 | 0 | 0 | 0 | 3 | 0 | 0 |

Шаг 6

0 1

2

t = 6

5 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **С** | G | G | G | G | B | W | G |
| **D** | 0 | 1 | I | 4 | 3 | I | 5 |
| **P** | N | 0 | 3 | 2 | 2 | N | N |
| **F** | 0 | 0 | 0 | 0 | 3 | 0 | 0 |

Шаг 7

0 1

2

t = 7

5 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **С** | G | G | G | G | B | G | G |
| **D** | 0 | 1 | I | 4 | 3 | 6 | 5 |
| **P** | N | 0 | 3 | 2 | 2 | 3 | N |
| **F** | 0 | 0 | 7 | 0 | 3 | 0 | 0 |

Шаг 8

0 1

2

t = 8

5 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **С** | G | G | B | G | B | G | G |
| **D** | 0 | 1 | 7 | 4 | 3 | 6 | 5 |
| **P** | N | 0 | 3 | 2 | 2 | 3 | 3 |
| **F** | 0 | 0 | 8 | 0 | 3 | 0 | 0 |

Шаг 9

0 1

2

t = 9

5 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **С** | G | G | B | G | B | B | G |
| **D** | 0 | 1 | 7 | 4 | 3 | 6 | 5 |
| **P** | N | 0 | 3 | 2 | 2 | 3 | 3 |
| **F** | 0 | 0 | 8 | 0 | 3 | 9 | 0 |

Шаг 10

0 1

2 3 4

t = 10

5 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **С** | G | G | B | G | B | B | B |
| **D** | 0 | 1 | 7 | 4 | 3 | 6 | 5 |
| **P** | N | 0 | 3 | 2 | 2 | 3 | 3 |
| **F** | 0 | 0 | 8 | 0 | 3 | 9 | 10 |

Шаг 11

0 1

2 3 4

t = 11

5 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **С** | G | G | B | B | B | B | B |
| **D** | 0 | 1 | 7 | 4 | 3 | 6 | 5 |
| **P** | N | 0 | 3 | 2 | 2 | 3 | 3 |
| **F** | 0 | 0 | 8 | 11 | 3 | 9 | 10 |

Шаг 12

0 1

2 3 4

t = 12

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **С** | G | B | B | B | B | B | B |
| **D** | 0 | 1 | 7 | 4 | 3 | 6 | 5 |
| **P** | N | 0 | 3 | 1 | 1 | 3 | 3 |
| **F** | 0 | 12 | 8 | 11 | 3 | 9 | 10 |

Шаг 13

0 1

2 3 4

t = 13

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **С** | B | B | B | B | B | B | B |
| **D** | 0 | 1 | 7 | 4 | 3 | 6 | 5 |
| **P** | N | 0 | 3 | 1 | 1 | 3 | 3 |
| **F** | 13 | 12 | 8 | 11 | 3 | 9 | 10 |

Обход в порядке:4256310

**3.Алгоритм топологической сортировки.**

По условию, граф имеет 7 вершин, пронумерованных начиная с нуля. В качестве стартовой вершины выбрана вершина с номером 0.

Задача топологической сортировки графа состоит в следующем: **указать такой линейный порядок на его вершинах, чтобы любое ребро вело от вершины с меньшим номером к вершине с большим номером.**

Топологическая сортировка – это процедура упорядочивания вершин ориентированного графа, не имеющего циклов.

При реализации топологической сортировки с помощью алгоритма поиска в глубину используется массив меток вершин, с помощью которого моделируется удаление вершин из графа и сохраняются новые номера вершин.

Шаг 1.

В качестве стартовой вершины выбираем вершину с номером 0. Окрашиваем ее в серый цвет.

0 1

2 3 4

5 6

Шаг 2.

Из нулевой идем в первую вершину, красим ее в серый цвет.

0 1

2 3 4

5 6

Шаг 3.

Из первой идем в третью вершину, красим ее в серый цвет.

0 1

2 3 4

5 6

Шаг 4.

Далее идём во вторую вершину и красим ее в серый цвет.

0 1

4

5 6

Шаг 5.

Из первой в четвёртую вершину, красим ее в серый цвет.

0 1

2 3 4

Шаг 6.

Переходим в пятую вершину и закрашиваем серым,так как 2 не имеет смежных вершин, красим ее в черный цвет и кладем в стек.

0 1

2 3 4

5 6

Шаг 7.

Возвращаемся в третью вершину и переходим в 6. Так как у 4 нет больше смежных вершин закрашиваем чёрным.

0 1

2 3 4

5 6

Шаг 8.

Возвращаемся в шестую вершину, красим ее в черный и кладем в стек.

0 1

2 3 4

5 6

Шаг 9.

Возвращаемся в пятую вершину, красим ее в черный и кладем в стек.

0 1

2 3 4

5 6

Шаг 10.

Возвращаемся в третью вершину, красим ее в черный и кладем в стек.

0 1

2 3 4

Шаг 11.

Переходим в первую вершину и красим ее в чёрный.

0 1

2 3 4

Шаг 12.

Возвращаемся в нулевую вершину, красим ее в черный и кладем в стек.

0 1

2 3 4

**В итоге получилась сортировка:** 2465310

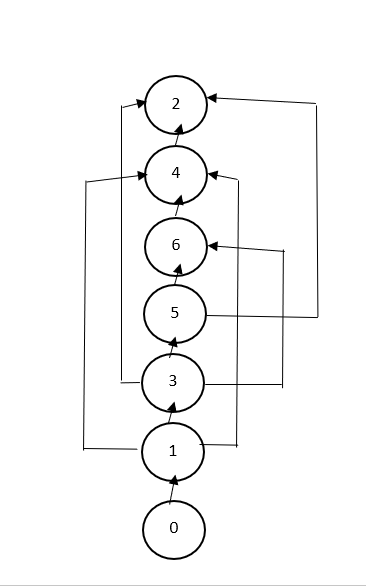


Рис.2 Сортировка графа

**Задание 2.** Осуществить программную реализацию алгоритмов на C++. Разработать структуры **AMatrix** и **АList**  для представления ориентированного графа матричным и списковым способом. Разработать функции преобразования из одного способа представления в другой. Разработать функцию **BFS** обхода вершин графа, используя метод поиска в ширину.

На рисунке 3 представлены разработанные структуры **AMatrix** и **АList**  для представления ориентированного графа матричным и списковым способом.

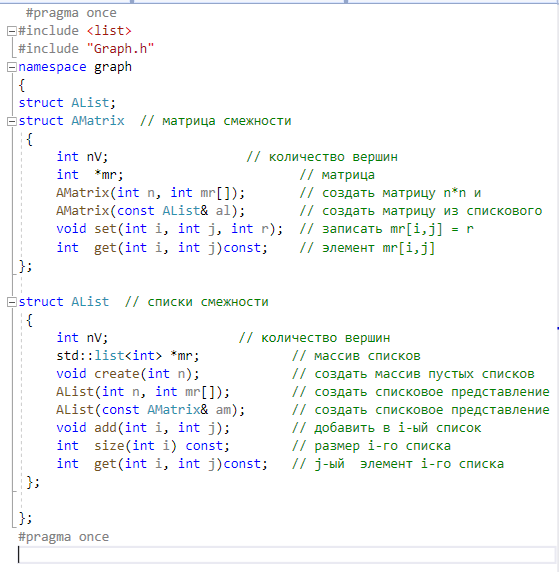


Рис.3 структуры AMatrix и АList

Разработать функции преобразования из одного способа представления в другой. Разработать функцию **BFS** обхода вершин графа, используя метод поиска в ширину.

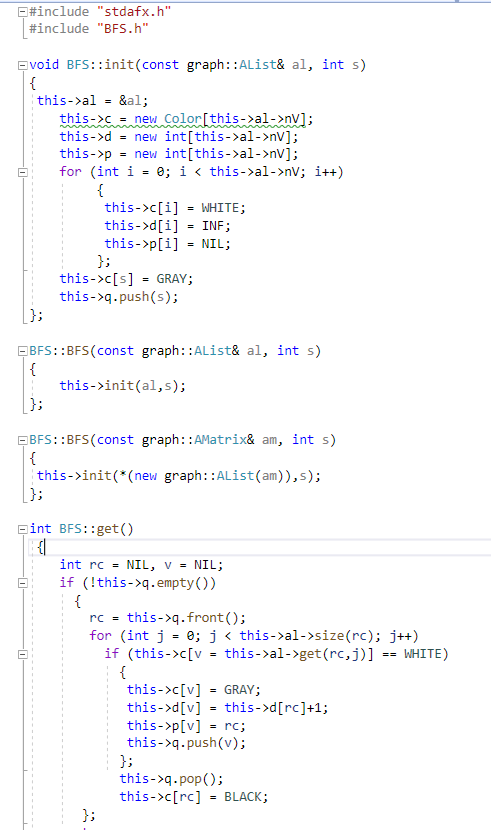


Рис.4 Функция **BFS** обхода вершин графа

**Задание 3-4.**  Разработать функцию **DFS** обхода вершин графа, используя метод поиска глубину. Продемонстрировать работу функции. Копии экрана вставить в отчет. Доработайте функцию **DFS**,для выполнения топологической сортировки графа. Продемонстрировать работу функции. Копии экрана вставить в отчет. Представлено на рисунке 5.

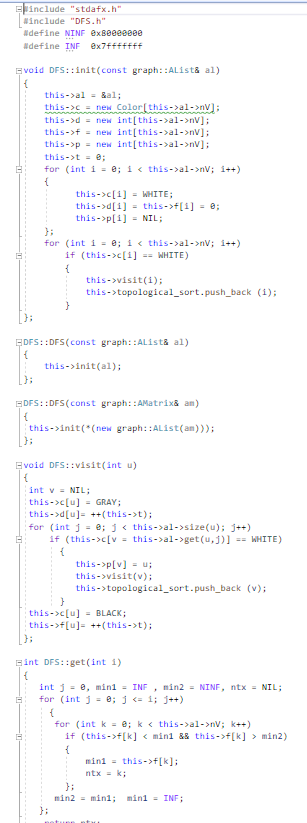


Рис.5 Функция **DFS**  обхода вершин графа

Программная реализация интерфейса представлена на рисунке 6.

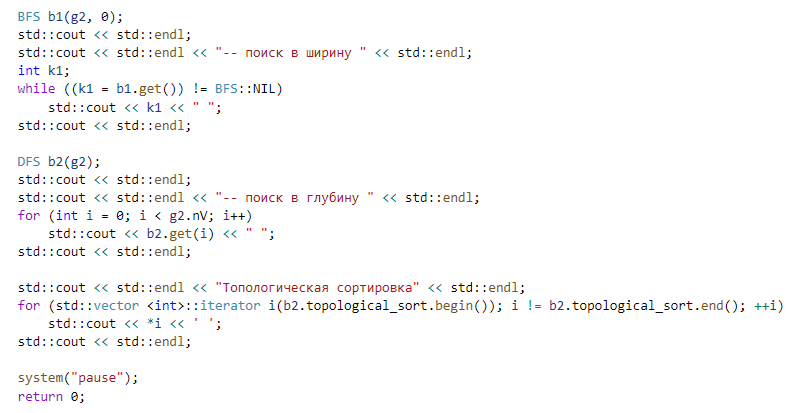
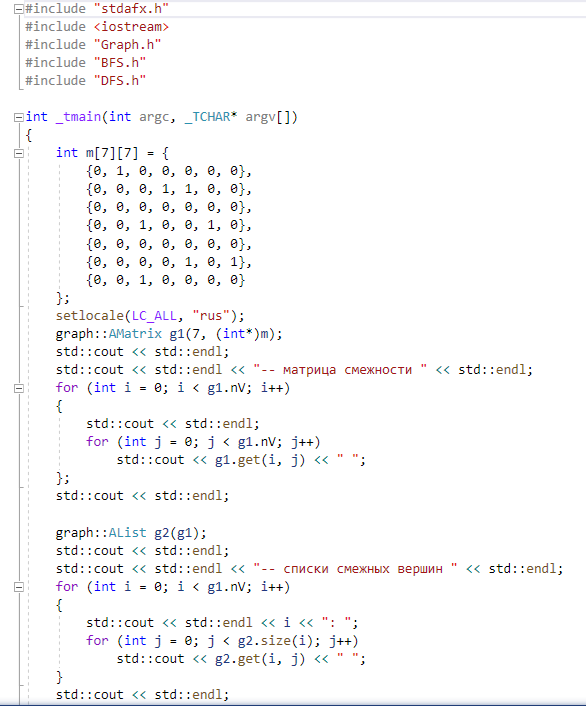


Рис.6 Программная реализация интерфейса

Продемонстрировать работу программы. Рисунок 7.

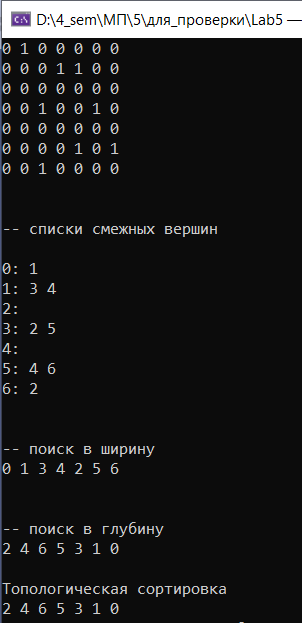


Рис.7 Программа демонстрирующая реализацию алгоритмов

**Вывод:** В ходе лабораторной работы были получены навыки в построении графов: алгоритмов поиска в ширину и глубину, алгоритма топологической сортировки графов.