Белорусский государственный технологический университет

Кафедра Информационных Систем и Технологий

**“Математическое программирование”**

**Отчет по лабораторной работе №5**

**Алгоритмы на графах**

**Вариант 5**

Выполнила: Деликатная М.М.

ЗФ 3 курс, 1 группа

Минск 2019

**Алгоритмы поиска в ширину**

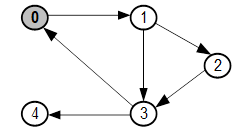
Исходный граф:



Осуществляем поиск в ширину. Некоторые обозначения: Q – очередь вершин, C – массив окраски вершин, P – массив предшествующих вершин, D – массив расстояний. Граф имеет 5 вершин, пронумерованных начиная с нуля. В качестве стартовой вершины выбрана вершина с номером 0.

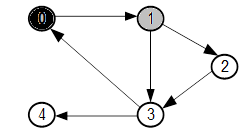
Шаг 1:

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



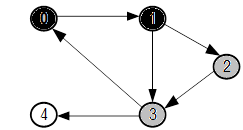
Шаг 2:

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



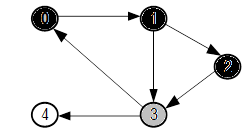
Шаг 3:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Q | 2 | 3 |  |  |  |
| C | B | B | G | G | W |
| D | 0 | 1 | 2 | 2 | I |
| P | N | 0 | 1 | 1 | N |



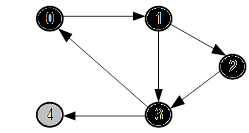
Шаг 4:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Q | 3 |  |  |  |  |
| C | B | B | B | G | W |
| D | 0 | 1 | 2 | 2 | I |
| P | N | 0 | 1 | 1 | N |



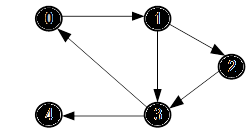
Шаг 5:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Q | 4 |  |  |  |  |
| C | B | B | B | B | G |
| D | 0 | 1 | 2 | 2 | 3 |
| P | N | 0 | 1 | 1 | 3 |

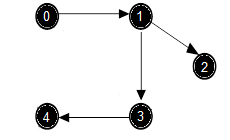


Шаг 6:

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



На 6 шаге обход дерева закончен. Все вершины черного цвета. В итоге получилось вот такое BFS – дерево:



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P | N | 0 | 1 | 1 | 3 |

**Алгоритмы поиска в глубину**

Исходный граф:

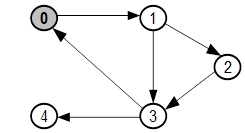


Осуществляем поиск в глубину. Назначение и размерность массивов: С – массив окраски вершин, Р – массив предшествующих вершин; такие же, как и в алгоритме поиска в ширину. В массив D для каждой вершины записывается время обнаружения (шаг окраски в серый цвет). Массив F предназначен для хранения времени фиксации (шага окраски в черный цвет) вершины. Кроме того используется переменная t, текущее значение которой – номер шага в алгоритме.

Шаг 1:

t=1

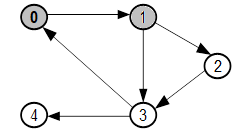
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | W | W | W | W |
| D | 1 | I | I | I | I |
| P | N | N | N | N | N |
| F | 0 | 0 | 0 | 0 | 0 |



Шаг 2:

t=2

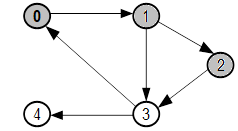
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | G | W | W | W |
| D | 1 | 2 | I | I | I |
| P | N | 0 | N | N | N |
| F | 0 | 0 | 0 | 0 | 0 |



Шаг 3:

t=3

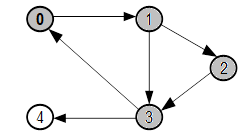
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | G | G | W | W |
| D | 1 | 2 | 3 | I | I |
| P | N | 0 | 1 | N | N |
| F | 0 | 0 | 0 | 0 | 0 |



Шаг 4:

t=4

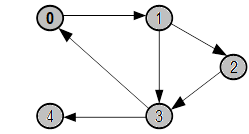
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | G | G | G | W |
| D | 1 | 2 | 3 | 4 | I |
| P | N | 0 | 1 | 2 | N |
| F | 0 | 0 | 0 | 0 | 0 |



Шаг 5:

t=5

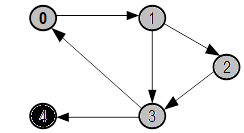
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | G | G | G | G |
| D | 1 | 2 | 3 | 4 | 5 |
| P | N | 0 | 1 | 2 | 3 |
| F | 0 | 0 | 0 | 0 | 0 |



Шаг 6:

t=6

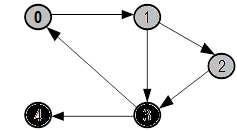
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | G | G | G | B |
| D | 1 | 2 | 3 | 4 | 5 |
| P | N | 0 | 1 | 2 | 3 |
| F | 0 | 0 | 0 | 0 | 6 |



Шаг 7:

t=7

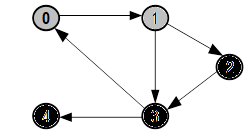
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | G | G | B | B |
| D | 1 | 2 | 3 | 4 | 5 |
| P | N | 0 | 1 | 2 | 3 |
| F | 0 | 0 | 0 | 7 | 6 |



Шаг 8:

t=8

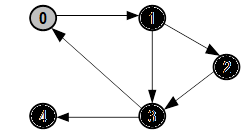
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | G | B | B | B |
| D | 1 | 2 | 3 | 4 | 5 |
| P | N | 0 | 1 | 2 | 3 |
| F | 0 | 0 | 8 | 7 | 6 |



Шаг 9:

t=9

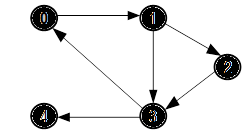
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | B | B | B | B |
| D | 1 | 2 | 3 | 4 | 5 |
| P | N | 0 | 1 | 2 | 3 |
| F | 0 | 9 | 8 | 7 | 6 |



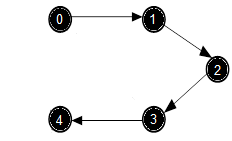
Шаг 10:

t=10

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



На 10 шаге алгоритм поиска в глубину окончен, все вершины окрашены в черный цвет. В конце получилось вот такое DFS – дерево:



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P | N | 0 | 1 | 2 | 3 |

**Алгоритм топологической сортировки**

Исходный граф:



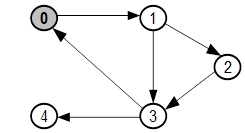
Топологическая сортировка – это процедура упорядочивания вершин ориентированного графа, не имеющего циклов.

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:

t=1

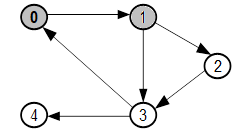
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | W | W | W | W |
| D | 1 | I | I | I | I |
| P | N | N | N | N | N |
| F | 0 | 0 | 0 | 0 | 0 |



Шаг 2:

t=2

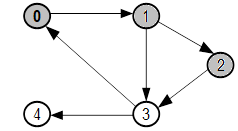
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | G | W | W | W |
| D | 1 | 2 | I | I | I |
| P | N | 0 | N | N | N |
| F | 0 | 0 | 0 | 0 | 0 |



Шаг 3:

t=3

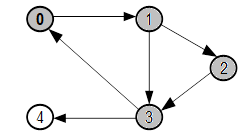
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | G | G | W | W |
| D | 1 | 2 | 3 | I | I |
| P | N | 0 | 1 | N | N |
| F | 0 | 0 | 0 | 0 | 0 |



Шаг 4:

t=4

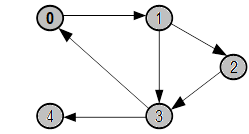
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | G | G | G | W |
| D | 1 | 2 | 3 | 4 | I |
| P | N | 0 | 1 | 2 | N |
| F | 0 | 0 | 0 | 0 | 0 |



Шаг 5:

t=5

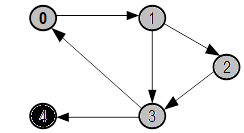
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | G | G | G | G |
| D | 1 | 2 | 3 | 4 | 5 |
| P | N | 0 | 1 | 2 | 3 |
| F | 0 | 0 | 0 | 0 | 0 |



Шаг 6:

t=6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | G | G | G | G | B |
| D | 1 | 2 | 3 | 4 | 5 |
| P | N | 0 | 1 | 2 | 3 |
| F | 0 | 0 | 0 | 0 | 6 |



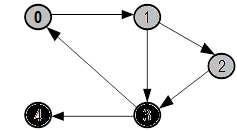
Из вершины номер 4 нет рёбер, идущих не в черные вершины. Возвращаемся к вершине номер 3. Красим вершину номер 4 в черный цвет и кладем ее в стек.

Стек: 4.

Шаг 7:

t=7

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



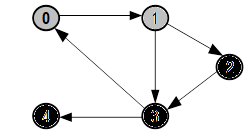
Из вершины номер 3 нет рёбер, идущих не в черные вершины. Возвращаемся к вершине номер 2. Красим вершину номер 3 в черный цвет и кладем ее в стек.

Стек: 3, 4.

Шаг 8:

t=8

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



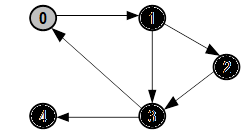
Из вершины номер 2 нет рёбер, идущих не в черные вершины. Возвращаемся к вершине номер 1. Красим вершину номер 2 в черный цвет и кладем ее в стек.

Стек: 2, 3, 4.

Шаг 9:

t=9

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



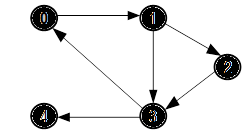
Из вершины номер 1 нет рёбер, идущих не в черные вершины. Возвращаемся к вершине номер 0. Красим вершину номер 1 в черный цвет и кладем ее в стек.

Стек: 1, 2, 3, 4.

Шаг 10:

t=10

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



Из вершины номер 0 нет рёбер, идущих не в черные вершины. Красим вершину в черный цвет и кладем ее в стек. Обход точек закончен.

Стек: 0, 1, 2, 3, 4.

Новые номера присваиваются в порядке вытаскивания из стека. Алгоритм топологической сортировки завершен. Граф отсортирован.

