Министерство образования Республики Беларусь

Учреждение образования “Белорусский государственный технологический университет”

Кафедра полиграфического оборудования и систем обработки информации

**Лабораторная работа №5**

**Алгоритмы на графах**

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

**Выполнил:** Грудинский Павел Владимирович

Минкск 2020

**Алгоритм поиска в ширину (BFS)**

Исходный граф:

Текущее состояние алгоритма хранится в следующих структурах памяти:

Q – очередь вершин,

C – массив окраски вершин,

D – массив расстояний,

P – массив предшествующих вершин.

По условию, 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: Извлекаем из очереди 0-ую вершину и закрашиваем в black (B) цвет; помещаем 1-ую вершину в очередь.

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

Шаг 3: Извлекаем 1-ую вершину, красим в B-цвет и помещаем 2, 3 вершины в очередь.

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

Шаг 4: Извлекаем 2-ую вершину, красим в B-цвет и помещаем 2, 3 вершины в очередь.

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

Шаг 5: Извлекаем вершину 3, закрашиваем ее и ищем ей смежные, т.е 4

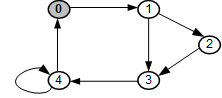
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Q | 4 |  |  |  |  |
| C | B | B | B | B | G |
| D | 0 | 1 | 2 | 2 | 3 |
| P | N | 0 | 1 | 1 | 2 |

Шаг 6: Последний шаг алгоритма. Извлекаем последний элемент из очереди, очередь пуста, все вершины окрашиваются в В-цвет. Извлекаем последнюю 4-ую вершину, проверяем на смежность с другими вершинами.

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

**Алгоритм поиска в глубину (DFS)**

Исходный граф:



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.

Кроме того, используется переменная t, текущее значение которой – номер шага алгоритма.

По условию, граф имеет 5 вершин, пронумерованных начиная с нуля. В качестве стартовой вершины выбрана вершина с номером 0.

Шаг 1.

t=1-стартовый шаг

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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. Ищем смежные вершины вершине 1. Это вершины 2 и 3. Выбираем наименьшую вершину. Окрашиваем 2 вершину в серый цвет.

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. Находим смежные вершины вершине 2. Это вершина 3.

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. Находим смежные вершины вершине 3. Это вершина 4.

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. Белых вершин, смежный с 4 вершиной, нет, т.о. мы погрузились в глубину. Закрашиваем 4 вершину в чёрный цвет.

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. На предыдущем и последующих шагах массивы D и P – не изменяются; заполняется массив F и изменяется массив C.

t = 7

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

Шаг 8.

t = 8

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

Шаг 9.

t = 9

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

Шаг 10.

t = 10

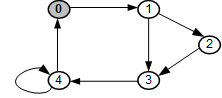
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C | B | B | B | B | B |
| D | 1 | 2 | 3 | 4 | 5 |
| P | N | 0 | 1 | 2 | 3 |
| F | 10 | 9 | 8 | 7 | 6 |

Ответ: DFS-дерево имеет вид:

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

**Алгоритм топологической сортировки**

Исходный граф:



По условию, граф имеет 5 вершин, пронумерованных начиная с нуля. В качестве стартовой вершины выбрана вершина с номером 0.

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.

Решение:

1/10

2/9

3/8

4/7

5/6

Все действия повторяются из алгоритма поиска в глубину.

Формируем очередь по мере окрашивания вершин в чёрный цвет: 0 1 2 3 4

В итоге получаем дерево:

Т.к. данный граф цикличен –

для него невозможно проставить

все «переходы». Проставляется

только переход с вершины

1 во 2 и в 3, что отображено

на рисунке.

Выполнено програмно:

