

Алгоритмы на графах

Топологическая сортировка поиском в глубину

*Югов Иван Олегович
МОУ Гимназия №10, г. Тверь*

Домашнее задание

Предприятие «Авто-2010» выпускает двигатели известных во всём мире автомобилей. Двигатель состоит ровно из n деталей, пронумерованных от 1 до n , при этом деталь с номером i изготавливается за p_i секунд. Специфика предприятия «Авто-2010» заключается в том, что там одновременно может изготавливаться лишь одна деталь двигателя. Для производства некоторых деталей необходимо иметь предварительно изготовленный набор других деталей.

Генеральный директор «Авто-2010» поставил перед предприятием амбициозную задачу — за наименьшее время изготовить деталь с номером 1, чтобы представить её на выставке.

Требуется написать программу, которая по заданным зависимостям порядка производства между деталями найдёт наименьшее время, за которое можно произвести деталь с номером 1.

Домашнее задание

Первая строка входного файла *details.in* содержит число n ($1 \leq n \leq 10\,000$) — количество деталей двигателя. Вторая строка содержит n натуральных чисел p_1, p_2, \dots, p_n , определяющих время изготовления каждой детали в секундах. Время для изготовления каждой детали не превосходит 10^9 секунд. Каждая из последующих n строк входного файла описывает характеристики производства деталей. Здесь i -я строка содержит число деталей k_i , которые требуются для производства детали с номером i , а также их номера. Сумма всех чисел k_i не превосходит 200000. Известно, что не существует циклических зависимостей в производстве деталей.

В первой строке выходного файла *details.out* должны содержаться два числа: минимальное время (в секундах), необходимое для скорейшего производства детали с номером 1 и число k деталей, которые необходимы для этого производства. Во второй строке требуется вывести через пробел k чисел — номера деталей в том порядке, в котором их следует производить для скорейшего производства детали с номером 1.

Ограничение по времени — 2 сек. Ограничение по памяти — 64 Мб.

Домашнее задание

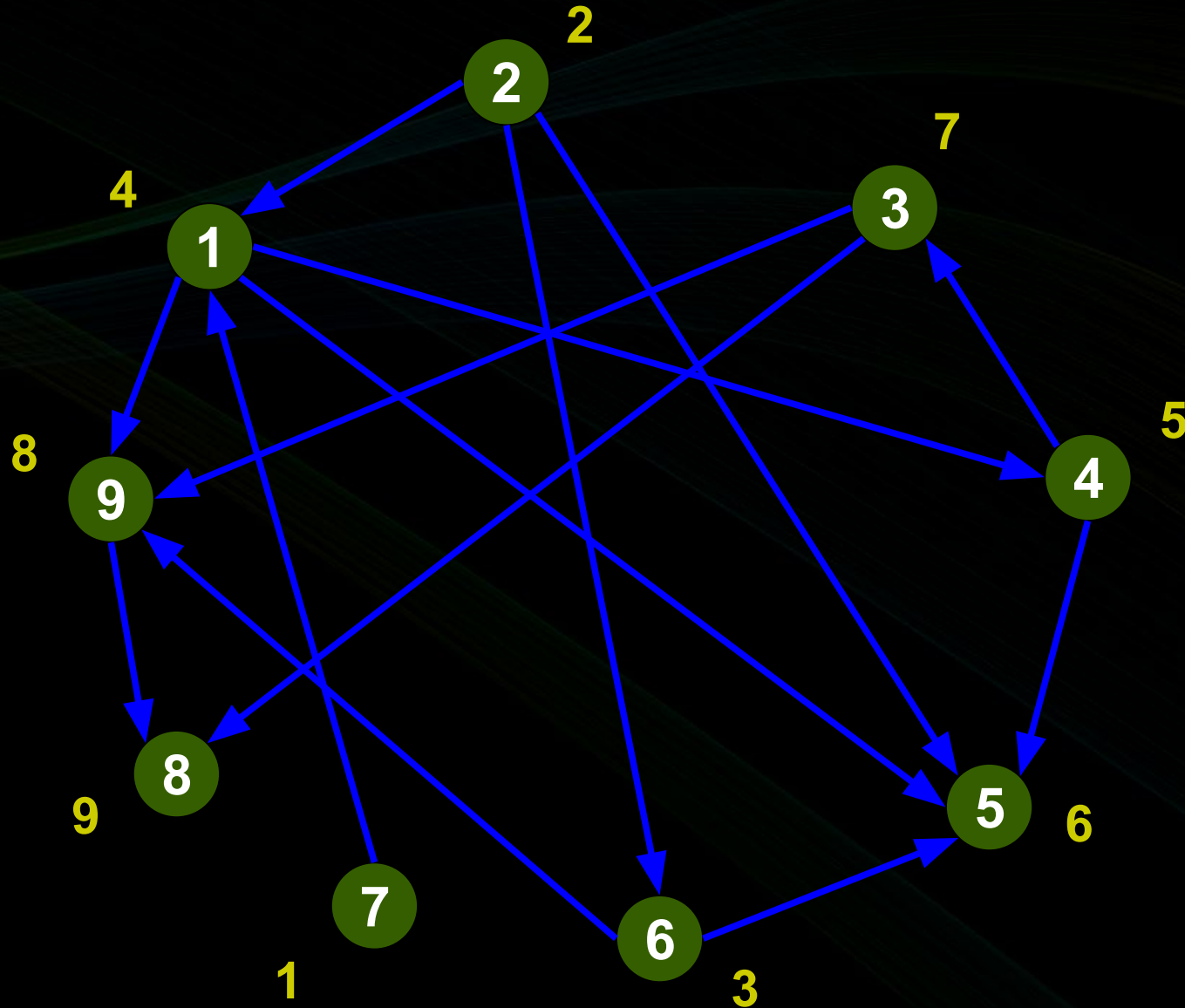
Топологическая сортировка

Для топологической сортировки можно использовать поиск в глубину.

Используем поиск в глубину от всех вершин со входящей степенью 0. Будем нумеровать вершины, из которых выходим, в обратном порядке.

При этом окажется, что граф отсортирован топологически.

Топологическая сортировка



Топологическая сортировка

Почему так происходит?

Не бывает рёбер из вершин, которые были покинуты раньше, в вершины, которые были покинуты позже.

Как это доказать?

Предположим, что это не так.

Пусть имеются вершины A и B , причём в процессе DFS вершина A была покинута раньше, чем вершина B . Пусть теперь существует ребро из A в B .

Топологическая сортировка

Рассмотрим момент времени, когда мы покидаем A (но ещё не покинули B).

Два случая:

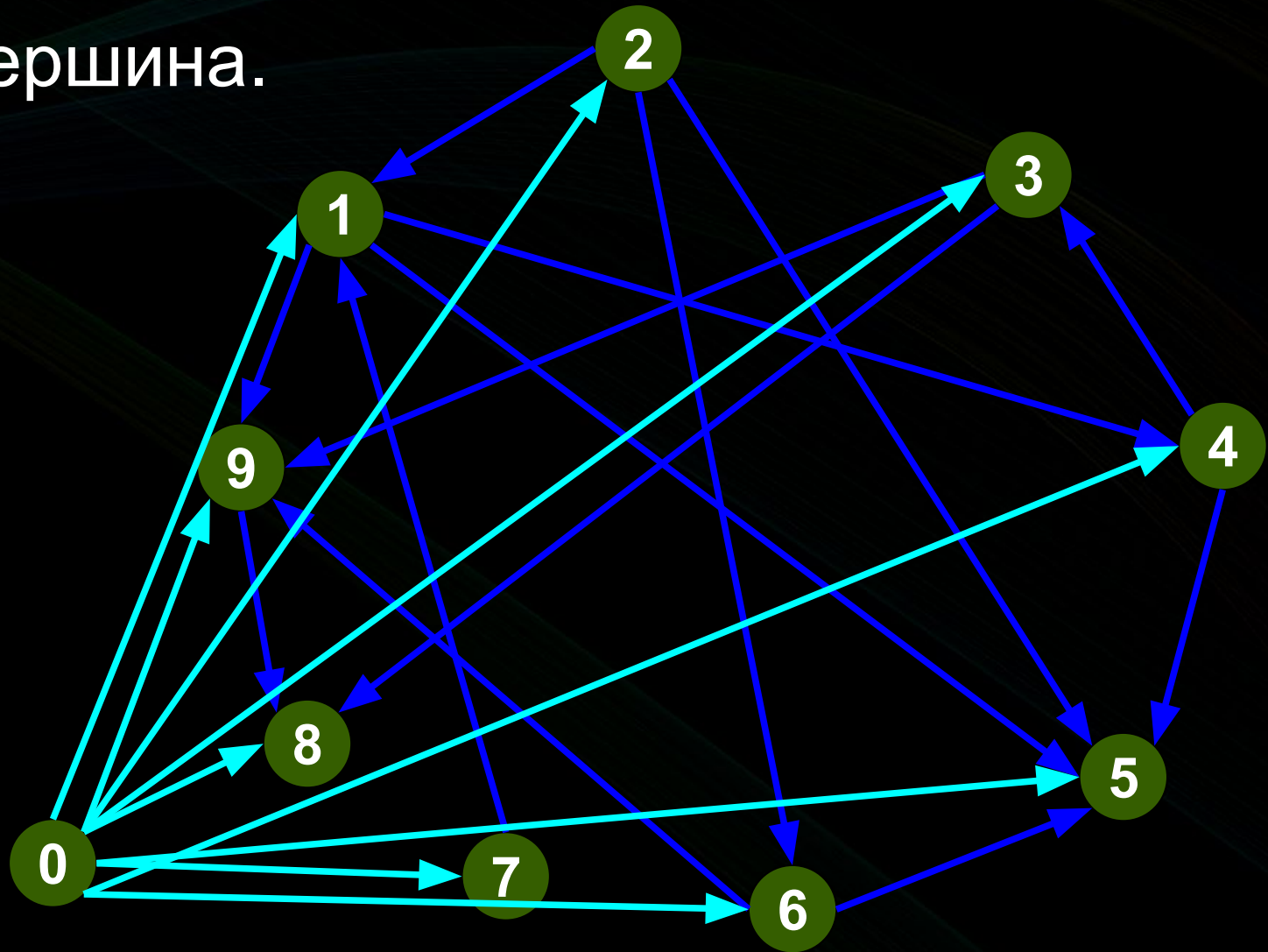
- 1) Покидаем A и ещё не посещали B .
- 2) Покидаем A и посетили B .

Однако:

- 1) Из чёрной вершины в белую не бывает рёбер.
- 2) Есть серый путь из B в A , что с ребром из A в B даёт цикл, но граф ациклический.

Топологическая сортировка

Альтернативное начало —
фиктивная вершина.



Топологическая сортировка

Если оказалось, что граф содержит циклы:

- как будет вести себя алгоритм топологической сортировки отсечением вершин?
- как будет вести себя алгоритм топологической сортировки поиском в глубину?

Домашнее задание

1. Какое максимальное количество рёбер может быть в ориентированном ациклическом графе с n вершинами?
2. Может ли быть так, что правильным результатом топологической сортировки графа оказывается любой порядок его вершин?
3. Решить задачу о производстве деталей с помощью DFS.
4. Как использовать топологическую сортировку для определения наличия циклов в графе?

Источники

Курс «Базовые алгоритмы для школьников»
(Станкевич А. С., Абакумов К. В., Мухачёва
М. А.)

«Интернет-университет информационных
технологий»

<http://www.intuit.ru/department/algorithms/basicalgos/>