



Алгоритмы и структуры данных

Лекция 9. Графы (разные задачи).

(с) Глухих Михаил Игоревич, glukhikh@mail.ru

Другие задачи на графе

- Топологическая сортировка
 - Упорядочение вершин ориентированного графа
- Построение минимального остовного дерева
 - Выделение минимального связного подграфа
- Поиск циклов в графе

Топологическая сортировка

- Определена на ориентированном графе без циклов (направленных)
- Выход: упорядоченный список вершин, такой, что для ребра $A \rightarrow B$ вершина A в списке всегда раньше B

Топологическая сортировка

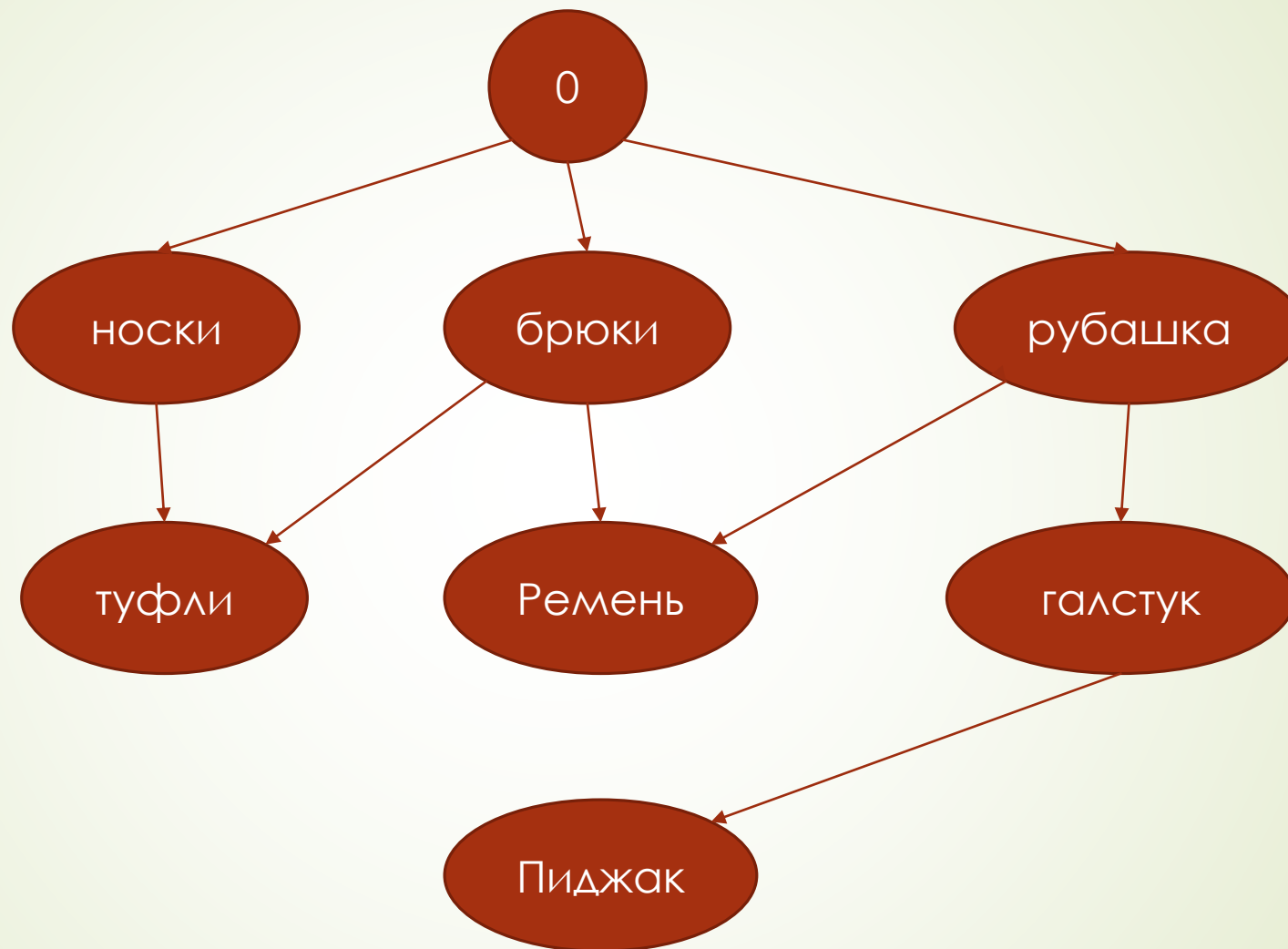
- Определена на ориентированном графе без циклов (направленных)
- Выход: упорядоченный список вершин, такой, что для ребра $A \rightarrow B$ вершина A в списке всегда раньше B
- Применяется, например, для упорядочения выполнения действий, некоторые из которых зависят от других (сборка в Gradle или через Makefile, пакеты программ вроде rpm и так далее)

Топологическая сортировка

- Определена на ориентированном графе без циклов (направленных)
- Выход: упорядоченный список вершин, такой, что для ребра $A \rightarrow B$ вершина A в списке всегда раньше B
- Задача одевания профессора Рассеянного
 - Рубашку (A) надо одевать раньше пиджака (B): $A \rightarrow B$
 - А брюки (C) раньше ботинок (D): $C \rightarrow D$

Топологическая сортировка

- Определена на ориентированном графе без циклов (направленных)
- Выход: упорядоченный список вершин, такой, что для ребра $A \rightarrow B$ вершина A в списке всегда раньше B
- Алгоритм Тарьяна:
 - Вводим (для удобства) вершину “0”, соединённую ребром со всеми вершинами графа
 - Обходим граф с помощью поиска в глубину, начиная с “0”
 - При **выходе** из вершины ставим её **в начало** итогового списка
 - Т.е.: в тот момент, когда поиск в глубину вершину окончательно покидает



Поиск циклов в графе

- ▶ Применение
 - ▶ Анализ программ
 - ▶ Анализ типов
 - ▶ ...

Поиск циклов в графе

- Применение
 - Анализ программ
 - Анализ типов
 - ...
- Алгоритм Тарьяна
 - Точно так же используем поиск в глубину
 - Если второй раз входим в уже посещенную вершину – найден цикл!

Минимальное остовное дерево

- Задача: оставить во взвешенном графе только часть рёбер так, чтобы все вершины по-прежнему были связаны и суммарный вес этих рёбер был минимальный

Минимальное остовное дерево

- Задача: оставить во взвешенном графе только часть рёбер так, чтобы все вершины по-прежнему были связаны и суммарный вес этих рёбер был минимальный
 - Вариант: число этих рёбер было минимально

Минимальное остовное дерево

- **Задача:** оставить во взвешенном графе только часть рёбер так, чтобы все вершины по-прежнему были связаны и суммарный вес этих рёбер был минимальный
 - **Вариант:** число этих рёбер было минимально
 - **Очевидное дополнение:** циклов в минимальном остовном дереве нет

Минимальное остовное дерево

- Задача: оставить во взвешенном графе только часть рёбер так, чтобы все вершины по-прежнему были связаны и суммарный вес этих рёбер был минимальный
- Применение: например, оптимизация компьютерной / дорожной сети

Минимальное остовное дерево

- Задача: оставить во взвешенном графе только часть рёбер так, чтобы все вершины по-прежнему были связаны и суммарный вес этих рёбер был минимальный
- Алгоритм Краскала
 - Взять пустое дерево
 - Пока возможно, добавлять в него ребро графа с минимальным возможным весом (из тех, которые не приводят к появлению цикла)

Поиск компонент связности

- Делаем BFS или DFS из произвольной вершины, обойдённые вершины запоминаем
- Если ещё остались вершины – повторяем процедуру, начиная с любой из оставшихся вершин
- И так далее...

Что-то ещё?

- ▶ Intentionally left blank...

Задачи по теории графов (1)

- Дано бинарное дерево (произвольное)
- По двум узлам найти ближайшего общего предка

Задачи по теории графов (1)

- Дано бинарное дерево (произвольное)
- По двум узлам найти ближайшего общего предка
- Идеи для решения:
 - В бинарном дереве поиска можно идти вверх, анализируя $> vS <$
 - В обычном бинарном дереве проще идти вниз, причём один узел всегда должен быть справа, другой слева

Задачи по теории графов (2)

- Даны два бинарных дерева
- Определить, является ли 2-е дерево поддеревом 1-го

Задачи по теории графов (2)

- ▶ Даны два бинарных дерева
- ▶ Определить, является ли 2-е дерево поддеревом 1-го
- ▶ Идеи для решения
 - ▶ ~ straight-forward: ищем в первом дереве корень второго, затем под ним ищем остальное поддерево

Задачи по теории графов (3)

- ▶ Дан связный граф
- ▶ Найти все «важные» рёбра или «мосты», исчезновение любого из которых приведёт к тому, что граф станет несвязным

Задачи по теории графов (3)

- Дан связный граф
- Найти все «важные» рёбра или «мосты», исчезновение любого из которых приведёт к тому, что граф станет несвязным
- Идеи для решения
 - Ни один из мостов не может входить в цикл! И наоборот!
 - Разметить на графе все циклы, остальное объявить мостами

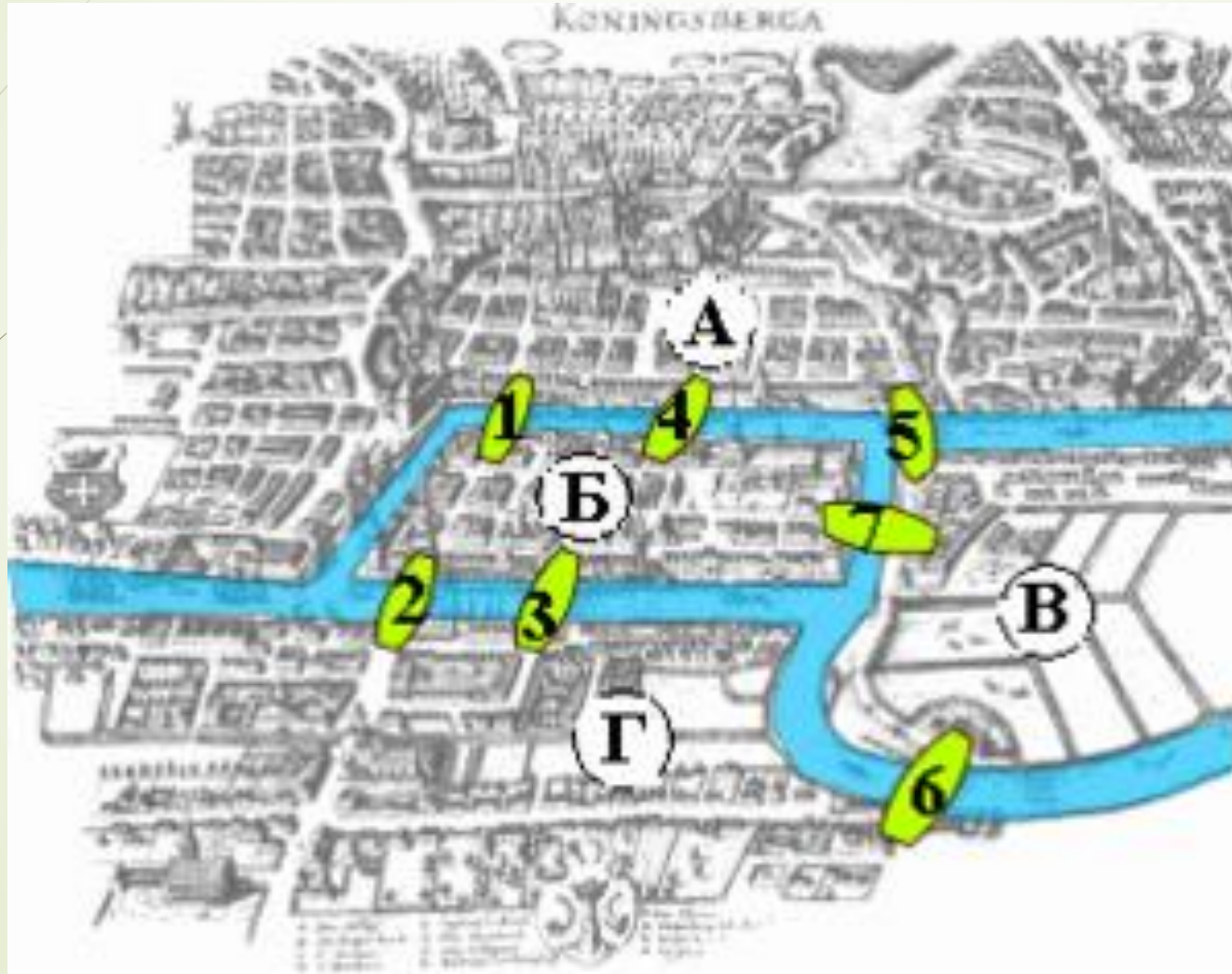
Задачи по теории графов (4)

- ▶ Эйлеров цикл – цикл, проходящий через все рёбра связного графа ровно по одному разу

Задачи по теории графов (4)

- Эйлеров цикл – цикл, проходящий через все рёбра связного графа ровно по одному разу
- Существует, если каждой вершине графа инцидентно чётное число рёбер
- Задача: найти Эйлеров цикл по графу

Задача по теории графов (4)



Задачи по теории графов (4)

- Эйлеров цикл – цикл, проходящий через все рёбра связного графа ровно по одному разу
- Существует, если каждой вершине графа инцидентно чётное число рёбер
- Задача: найти Эйлеров цикл по графу

- Идеи для решения:
 - Найти цикл, запомнить и выкинуть из графа рёбра этого цикла.
 - Повторять, пока рёбра не кончатся. Соединить все циклы в один.

Задачи по теории графов (5)

- Дана матрица из букв
- Найти в ней все имеющиеся слова (разрешается переходить от буквы к соседней по горизонтали / вертикали с любым числом поворотов)

И **Т** Ы Н

К **Р** **А** Н

А К **В** **А**

Задачи по теории графов (5)

- Дана матрица из букв
- Найти в ней все имеющиеся слова (разрешается переходить от буквы к соседней по горизонтали / вертикали с любым числом поворотов)
- Идеи для решения
 - Поиск в глубину с проверкой текущего слова по словарю
 - Возможно использование префиксного дерева для поиска по словарю