Лекция 8

0371 Кузнецова Елизавета

26 October 2021

Длины путей в графе

Определение 1. *Длина путей в графе - количество ребер в пути.*

Пример:

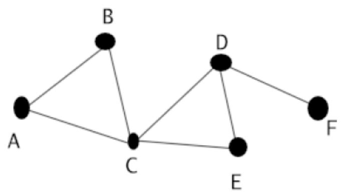

Рис. 1: Пример

ABCDF - путь длины 4 (4 ребра) от A до F
ACEDF - длина 4
ACDF - длина 3
ABCEDF - длина 5

Определение 2. *Расстояние между вершинами - минимальная длина пути между вершинами или $+\infty$, если пути нет*

Обозначение:$d(X, Y)$ - расстояние от X до Y
Пример: $d(A, F) = 3$

Определение 3. *Диаметр графа - максимальное расстояние между вершинами графа*

Пример:

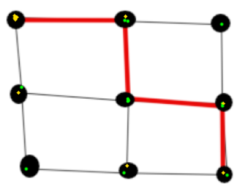

Рис. 2: Диаметр графа - 4

Определение 4. *Для каждой вершины графа* $G = (V, E)$ *можно посчитать max расстояние до других вершин*
$(v) = max\{d(V, s)|s \in V\}$
Радиус: $r(G) = min\{d(V, s)|s \in V\}$
Вершины, на которых достигается минимум - это центр

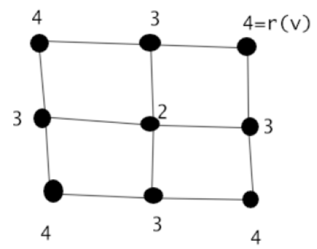

Рис. 3: Диаметр графа - 2

Центров может быть несколько

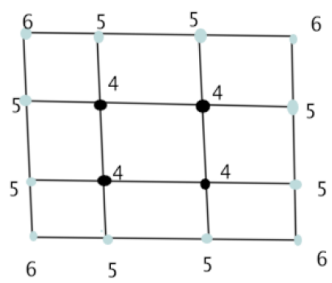

Рис. 4: 4 центра

Утверждение: В $G = (V, E)$ $d(G) \leq 2r(G)$

Доказательство. Пусть c - центр графа $u, v \in V$, $d(c, u) \leq r$

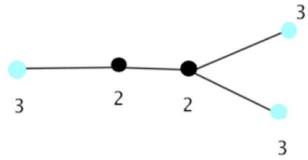

Рис. 5: дерево

$\Rightarrow d(u,v) \leq 2r \Rightarrow d(G) = maxd(u,v) \leq 2r$

$\square$

Утверждение: В дереве ≤ 2 центров
Пусть их 3

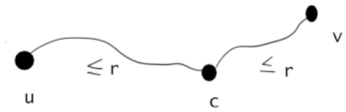

Рис. 6:

Построим пути между c_1 В c_2, потом c_2 и c_3 (в дереве ровно 1 путь между вершинами).

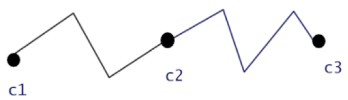

Рис. 7:

$r(c_0) < r(c_1) = r(c_2) = r(c_3) = r(G) = r$
Замечание: Будем дальше иногда использовать ориентированные графы $G = (V, E)$
Ребра в ориентированном графе иногда называют дугами
$E \in \{(u, v)$ - упорядоченная пара $\}$
Пример:

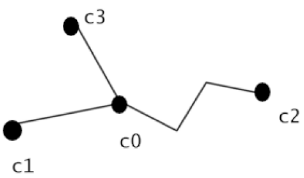

Рис. 8: c_0 - вершина развилки

Замечание: У ребер будут веса
$G = (V, E)$ вес - это $f : E \to \mathbf{R}$

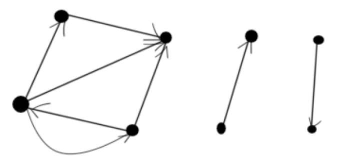

Рис. 9: Ориентированные графы

Расстояние на графе с весами считается как минимум суммы весов по всем путям

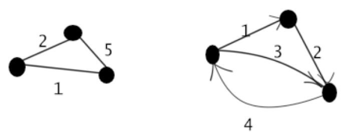

Рис. 10: Графы с весами

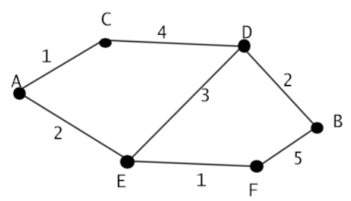

Рис. 11: Графы с весами

$d(A, B) = ?$
$d(ACDB) = 1 + 4 + 2 = 7$
$d(ACDEFB) = 1 + 4 + 3 + 1 + 5 = 14$
$d(AEFB) = 2 + 1 + 5 = 8$
$d(AEDB) = 2 + 3 + 2 = 7$
$min = 7$
$\Rightarrow d(A, B) = 7$
Замечание: Расстояние во взвешенном графе не всегда существует

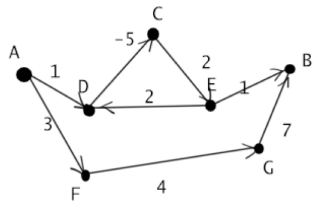

Рис. 12:

$d(F,G) = 4$
$d(G,F) = +\infty$
$d(A,B) = ?$
$d(ADCEB) = 1 - 5 + 2 + 1 = -1$
$d(ADCEDCEB) = 1 = 5 + 2 + 2 = 5 + 2 + 1 = -2$ и т.д.
$min = -\infty$
Утверждение: В графе есть все расстояния $\Leftrightarrow$ в графе нет цикла отрицательной длины

Доказательство. Если есть цикл $< 0 \Rightarrow \forall$ две вершины этого расстояния не имеют расстояния или $-\infty$
Если нет расстояния, т.е. для u, v есть пути сколь угодно маленькие. Пусть есть путь длиннее $n = |V|$ ребер $\Rightarrow$ повтор вершин в пути - это будет отрицательный цикл

$\square$

Как хранятся графы в компьютере

1. Матрица смежности - таблица вершины х вершины $a(i,j) = \{$ 0, если нет ребра
1, если есть ребро

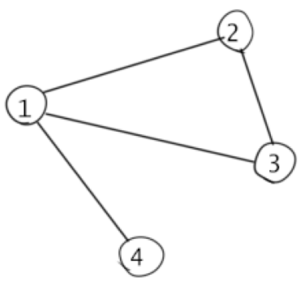

Рис. 13:

	1	2	3	4
1	0	1	1	1
2	1	0	1	0
3	1	1	0	0
4	1	0	0	0

Симметрична для неориентированного графа
Для графов с весами: $a(i, j)$ = вес ребра ij или $+\infty$, если нет

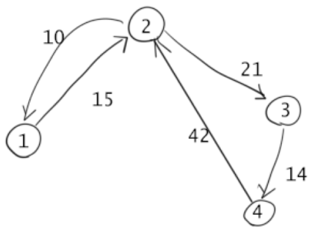

Рис. 14:

	1	2	3	4
1	$+\infty$	15	$+\infty$	$+\infty$
2	10	$+\infty$	21	$+\infty$
3	$+\infty$	$+\infty$	$+\infty$	14
4	$+\infty$	42	$+\infty$	$+\infty$

2. Списки смежности: для каждой вершины хранят список соседей
1: 2(15)
2: 1(10), 3(21)
3: 4(14)
4: 2(42)
Память $\approx |E|$ - количество ребер
3. Неявные способы: умеем вычислять всех соседей $\forall$ вершины.
Пример: Зададим обход конем шахматной доски

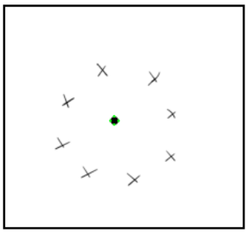

Рис. 15:

Граф: вершины - клетки 64 штуки
Ребра - вершины через ход коня
Можно для любой клетки посчитать, куда можно попасть. Задача: даны 2 вершины u, v, найти $d(u, v)$ и путь, на котором достигается это расстояние.
Замечание: оказывается, что найти путь от u До v это то же самое, что искать путь от u до всех вершин.
Алгоритм Форда-Беллмена
Дано $G = (v, E), u \in V$, найти расстояние $d(u, v)$ для $\forall v \in V$
Будем писать $d(V) = d(u, v)$, т.к. u не меняется.
Будем хранить в массиве d текущие найденные расстояния.
В началеЖ $d(u) = 0$, $d(v) = +\infty$, если v==u
Релаксация ребра $e = (v_1 v_2)$
Если $d(v_1) + f(v_1 v_2) < d(v_2) \Rightarrow d(v_2) = d(v_1) + f(v_1 v_2)$
Алгоритм: повторяем $n-1$ раз. Перебираем все ребра е и каждое релаксируем.
В неориентированном графе ребро в две сторон, т.е. две релаксации на ребро.
Пример:

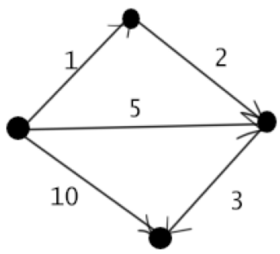

Рис. 16:

	A	B	C	D
	0	$+\infty$	$+\infty$	$+\infty$
AB	0	1	$+\infty$	$+\infty$
AC	0	1	5	$+\infty$
AD	0	1	5	10
AB	0	1	3	10
CD	0	1	3	6

Ответ:
$d(A) = 0$
$d(B) = 1$
$d(C) = 3$
$d(D) = 6$
Время работы $\approx |V| * |E| \leq |V|^3$