

## 93. Определение дерева, его свойства (б/д). Определение диаметра дерева. Алгоритм поиска диаметра в дереве.

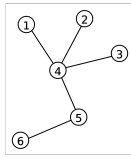


Рис. 1: Пример дерева

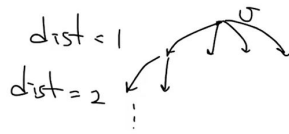


Рис. 2: DFS в дереве

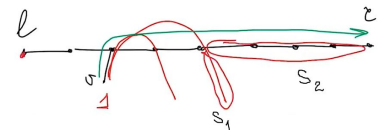


Рис. 3: Корректность поиска диаметра

$G$  - **дерево**, если  $G$  - неориентированный связный граф без циклов.

$E(G)$  - рёбра графа/дерева (edges),  $V(G)$  - вершины дерева.

Свойства деревьев:

1.  $|E(G)| = |V(G)| - 1$ ; количество рёбер на один меньше количества вершин дерева.
2.  $\forall u, v \exists!$  простой путь из  $u$  в  $v$ .

Пусть  $G$  - неориентированный невзвешенный граф.  $dist(s, t)$  - кратчайшее расстояние между вершинами  $s$  и  $t$ . **Диаметр  $G$**  - такой путь между вершинами  $u$  и  $v$ , что  $dist(u, v) = \max_{s, t} dist(s, t)$ . Обозначение:  $diam(G)$ .

Тезис: многие задачи на графах являются очень трудными/пр-трудными, а на деревьях решаются быстро. Пример 1: Поиск максимальной клики. Пример 2: Нахождение диаметра. Произвольный граф: Флойд - за  $O(n^3)$ . В дереве ищется за  $O(n)$ .

### Алгоритм поиска диаметра в дереве:

Поиск самой далёкой вершины: "подвесим" дерево за  $v$ , дальше запускаем dfs. В силу вида дерева (см. рис. 2) мы не можем подняться на другой слой/попасть в вершину, которую мы уже посещали. Победа! -  $x$  - вершина с самого низкого уровня.

- 1) Пусть  $v$  - произвольная вершина. Найдём  $l$  - самую далёкую вершину от  $v$ .
- 2) Теперь  $g$  - самая далёкая вершина от  $l$ . Тогда  $(l, g)$  - это диаметр.

Асимптотика:  $O(n)$

Корректность:

Посмотрим на рисунок три. Мы выбрали максимальный путь. По сути мы к каждой вершине подвесили своё дерево. Однако максимальные глубины подвешенных поддеревьев меняются в следующей последовательности:  $0, 1, 2, \dots, 2, 1, 0$  (так как иначе можно найти более длинный путь). Получается, для нашей произвольной  $v$   $s_1 \leq s_2$ . Тогда самый длинный путь из  $v$  имеет ту же длину, что путь от  $v$  до  $g$  (или до  $l$ , надо выбрать максимум). Но тогда найденная вершина находится от  $l$  на таком же расстоянии, как и  $g$ . Таким образом, мы нашли какой-то диаметр (не обязательно именно тот, который мы выбрали в начале, когда подвешивали деревья к вершинам).

**Альтернативный алгоритм:** подвесим дерево за произвольную  $v$ , среди всех поддеревьев  $v$  выбираем самые глубокие, вершины на низших слоях этих поддеревьев будут искомыми концами диаметра.

## 94. Определение центроида в дереве. Алгоритм поиска центроида в дереве. Лемма о количестве центроидов (б/д).

Пусть  $G$  - дерево,  $v \in V(G)$ . Тогда  $v$  - **центроид**, если после удаления  $v$  дерево распадается на несколько компонент, размером не более  $\frac{n}{2}$



Рис. 4: Пример центроидов



Рис. 5: Алгоритм поиска центроида: шаг 1

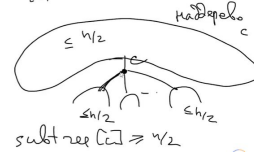


Рис. 6: Наддерево: шаг 2

**Утверждение 1.** В любом дереве есть центроид.

**Утверждение 2 (б/д).** В любом дереве либо один центроид, либо центроидов два и они соединены ребром (см. рис. 4)

**Алгоритм поиска центроида** за  $O(n)$ .

1. Пусть  $r$  - произвольный корень;  $\text{subtree}[v]$  = размер поддерева вершины  $v$ , включая  $v$  - это обычная динамика, dfs;  $\text{subtree}[r] = n$ .

1.5. Цель: найти такую вершину  $s$ , размеры всех поддеревьев детей которой не больше  $n/2$ , но  $\text{subtree}[s] \geq \frac{n}{2}$ . (Наддерево также станет одним из деревьев, образованных после удаления вершины; поддерево + наддерево = всё дерево)

P.S. Дабы избавиться от бед с округлениями, лучше сравнивать не так, как записано выше, а  $2\text{subtree}[s] \geq n$ .

2. Спускаемся от корня вниз к ребёнку с размером поддерева  $\text{subtree} \cdot 2 \geq n$ , пока можем. Как только такого ребёнка нет - вершина, где мы закончились, - центроид.