

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ПЪТИЩА

Задачата се свежда до това да намерим диаметрите на поддървета на началното дърво. Диаметър на граф е най-дългия път между два върха в графа.

Да разгледаме алгоритъм за намиране на диаметър в дърво:

1. Започваме от произволен връх V .
2. Намираме най-отдалечения от V връх, W (това може да се направи чрез линейно обхождане на дървото). Ако има повече от един кандидат за W , избираме произволен такъв.
3. Намираме най-отдалечения от W връх, U .
4. Диаметъра на графа е пътя между двата върха W и U .

Очевидно, този алгоритъм отнема линейно време.

Доказването на вярността на алгоритъма оставяме за упражнение (разгледайте диаметър AB в дървото, и се опитайте да покажете, че W съвпада или с A или с B . Достатъчно е да се разгледат пътищата AB и VW , и ребрата от дървото, които свързват двата пътя (може пътищата да се пресичат))

Използвайки директно този алгоритъм, ще получим решение $O(N^2)$ в зависимост от имплементацията, може да очакваме да получим точки за първите 2-3 подзадачи.

За да получим по-добро решение ще отговаряме на заявките в обратен ред. Вместо да премахваме ребра, ще добавяме ребра и ще смятаме диаметрите на новополучените дървета.

На всяка стъпка имаме две дървета T_1 с диаметър A_1B_1 , и T_2 с диаметър A_2B_2 . Добавяме ребро $E = (V_1, V_2)$, което ще свърже двете дървета. Да имитираме линейния алгоритъм за диаметър в дърво.

1. Започваме от V_1 .
2. Най-отдалечения връх от V_1 ще бъде някой от върховете A_1, B_1, A_2 или B_2 , тъй като най-далечния връх ще бъде край на диаметър в някое от дърветата T_1 или T_2 . Без ограничение може да смятаме, че най-отдалечения връх от V_1 е A_1 .
3. Най-отдалечения връх от A_1 ще бъде някой от B_1 (ако се намира в T_1), или A_2 или B_2 , ако най-далечния връх се намира в T_2 (защото един от двата върха е най-отдалечен от V_2 . Ако искаме да стигнем до T_2 трябва да минем по реброто (V_1, V_2)).

За да изпълним този алгоритъм, трябва $O(N)$ пъти да отговорим на запитване: колко е разстоянието между върховете V, U в началното дърво. Това може да се направи чрез

намиране на LCA (най-близък общ предшественик) в дърво.

(<https://www.topcoder.com/community/data-science/data-science-tutorials/range-minimum-query-and-lowest-common-ancestor/>) Решенията за 100 точки би трябвало да имат

сложност $O(N)$ или $O(N * \log N)$.

Автор: Йордан Чапъров