XLII НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА
Областен кръг, 14 февруари 2026 г.
Група C – 7, 8 клас

Задача C?. СЪОБЩЕНИЕ (Анализ на решението) ⏳0.2 сек. 💾 256 MB

За улеснение вместо за хора ще говорим за върхове в дърво.

Първа подзадача

За тази подзадача е предвидено решение с пълно изчерпване. Един възможен подход е да се фиксира начален връх и да се гледат всички наредби на върхове, съответсващи на реда, в който върховете получават съобщението, като за всяка се определи времето, нужно съобщението да се препрати до всичките върхове.

Втора подзадача

Тук е гарантирано, че дървото е звезда. Нека u е единственият връх, който има степен поне 2. Лесно се вижда, че е оптимално да започнем от u. Тогава отговорът е $\frac{N-1}{K}$.

Трета подзадача

Тук е гарантирано, че дървото е пръчка. Лесно се вижда, че е оптимално да започнем от центъра на пръчката (върхът, намиращ се в средата, или, ако има два - който да е от двата). Разграничават се случаи според K и четността на N, като за всеки от тях лесно се извежда формула.

Четвърта подзадача

Тук е гарантирано, че може даден връх за един ден да препрати съобщението до всичките си съседи. За целта търсим връх, в който, ако коренуваме дървото, най-дългият път от корена до някой връх, иначе казано дълбочината на дървото, да е минимален. Интуитивно е да разгледаме най-дългия път в самото дърво - диаметъра, като нека неговата дължина е d. Наистина, ако приложим логиката от трета подзадача върху диаметъра, бихме получили, че оптимален избор за корен е средата на диаметъра (центъра на дървото) и отговорът е $\left\lceil \frac{d}{2} \right\rceil$. От друга страна, можем да се убедим, че това е така, защото ако съществува връх, за който всеки връх е на разстояние най-много $\left\lceil \frac{d}{2} \right\rceil - 1$ от него, то най-дългият път в дървото би бил най-много $2 \left\lceil d \right\rceil 2 - 2 < d$, което влиза в противоречие с факта, че диаметърът е най-дългият път.

Пета и шеста подзадача

Нека сме фиксирали от кой връх започва препращането и сме коренували дървото в този връх. Забелязваме, че ако даден връх получи съобщението, то родителят му вече е получил съобщението и всъщност остава на този връх да препраща само на свои деца. Това ни довежда до мисълта, че може да използваме динамично програмиране. Нека $f(u)$ е минималното необходимо време да се препрати съобщението на поддървото на u, считано от момента, в който u е получил съобщението. За целта нека разполагаме с $f(u_1)$, …, $f(u_k)$ - отговорите за поддърветата на децата на u. Интуитивно бихме пращали на поддървета, на които ще им отнеме повече време, по-рано, и на такива, на които ще им отнеме по-малко време, по-късно. Действително, ако сортираме $f(u_i)$ в намаляващ ред, се оказва, че е оптимално съобщенията да се пращат в този ред. Можем да се убедим, като допуснем, че има два съседни върха в подредбата u_j и u_{j+1}, за които е вярно, че $f(u_j) < f(u_{j+1})$. Тогава лесно се вжида, че разменяйки двата върха в подредбата, отговорът не нараства.

Седма подзадача

Единственият минус на горното решение е, че е бавно, защото трябва да коренува дървото във всеки възможен връх. За целта ще използваме оптимизация, известна като rerooting. Нека коренуваме дървото във връх 1 и изчислим $f(u)$ като в горните подзадачи. Ще се стремим по същия начин да изчислим отговора за произволен връх u като корен, но без да коренуваме дървото в него. Всъщност да забележим, че всички ребра, излизащи от даден връх, водят до поддървета на децата на u, с изключение на най-много едно, което води до родителя на u. Но ако ни е налична подредбата, за която сме изчислили отговора за родителя на u, то можем да изчислим и отговора за родителя на u, като в този списък не включваме u. Това например се имплементира чрез префикс и суфикс по цялата подредба за даден връх. Тогава веднага се вижда, че можем да изчислим и отговора за u.

Автор: Кирил Зулямски