Анализ на задача tree2

Решения, условие, анализ: Димитър Шапатов; Решения, анализ: Кирил Зашев

Подзадача 1

Най-голямата подсказка тук за състезателите трябва да са ограниченията. Те са същите, като от задача Magic Trick от IOI 2025 Practice Session. За тези, които не знаят тази задача – ☹. В задачата трябва да се слагат максимален брой -1 на пермутация така, че след това да се декодира успешно (напомня ми на една друга задача, но това е отделен въпрос...) За да се реши задачата, трябва да се приложи теоремата на Erdős-Szekeres. Тя гласи, че всяка редица от $N \geq (a-1)(b-1)+1$ различни реални числа със сигурност съдържа растяща подредица с дължина a, или намаляваща подредица с дължина b. Това е ключовото наблюдение в задачата, но възниква проблем –в `decode` не знаем дали сме взели растяща или намаляща подредица. Този проблем можем да решим лесно. Знаем, че ако най-дългата растяща е с размер под $\lfloor\sqrt{N}\rfloor$, то най-дългата намаляваща е с размер поне $\lfloor\sqrt{N}\rfloor+1$, следователно ако най-дългата растяща подредица е с размер поне $\lfloor\sqrt{N}\rfloor$, взимаме първите ѝ $\lfloor\sqrt{N}\rfloor$ елемента, а в противен случай взимаме $\lfloor\sqrt{N}\rfloor+1$ елемента от най-дългата намаляваща подредица. Така различаваме кое от двете сме взели, в зависимост от това дали броят -1 е $\lfloor\sqrt{N}\rfloor$ или $\lfloor\sqrt{N}\rfloor+1$.

Как обаче това ни помага в тази задача? Оказва се, че можем да си сведем нашата задача до задача за пермутация, ако направим ойлеров тур на дървото, като запазваме върховете само, когато влизат. В декодирането няма да има проблем да се върне същия ойлеров тур от кодирането, защото ребрата се подават в същия ред, в който са били подадени на `encode` – така списъците на съседите се построяват в същия ред и обхождането е идентично. Различните премахнати върхове са дадени като различни отрицателни числа, така че това също не е проблем – можем да ги различаваме. Също нямаме проблеми с корена, тъй като винаги е в 1 или -1 и просто можем да използваме това, което имаме от двете.

Подзадача 2

По-голямото ограничение просто ни затормозява с това да направим бързия начин за намиране на най-дългата растяща подредица и най-дългата намаляваща редица чрез сегментно дърво. Тук имплементацията е още по-досадна.