Анализ на задача cactus

Наивни решения

Някои наивни решения включват идеи, които директно подават списъка на ребра / съседство на графа, без да се възползват от кактусовата му структура. Те получават между 10 и 20 точки.

Кодиране на дърво

Ще отделим малко внимание на решение за друга задача, което няма да доведе до точки на тази, но ще е основа за всички решения нататък: При дадено номерирано дърво с n върха, индексирани с числата от 0 до $n-1$, компресирайте го и го разкомпресирайте.

Пускаме dfs по дървото, който за всеки връх (освен корена) ще определи някакъв родител. Тези родители ще определят перумтация на $\{0, 1, \cdots, (n-1)\}$ (ще си затворим очите за единия липсващ елемент). Наивната идея да ги компресираме в двоична бройна система, заделяйки 14 бита за всяко число всъщност е и оптимална. Искаме да разграничим $n!$ обекта, асоциирайки ги с двоични низове – те трябва да имат дължина $\log n! = O(n \log n)$, което е точно нашия резултат.

Кодиране на кактус

Следвайки дефинициите дървото е граф без цикли, а кактусът е граф, в който всяко ребро е включено в най-много един цикъл. Миналият абзац за кодиране на дърво ни мотивира да построим покриващо дърво на графа ни чрез dfs. Сега ребрата ще се разделят на два типа – задни ребра (back edges) и дървесни ребра (tree edges). Ще отбележим още сега, че задните ребра първо свързват връх с негов прародител и второ задават циклите в графа (упражнение за читателя).

Задните ребра имат допълнителни свойства конкретно в кактусите, от които силно ще се възползваме. Първо, от всеки връх "излиза" по един back edge (в противен случай този връх и родителя му задават ребро, което ще участва в два цикъла, противоречие). Второ, задните ребра не се "пресичат" по какъвто и да е начин (допускането отново ще доведе до противоречие с дефиницията).

От тук можем да получим едно решение, което ще донесе ≈ 70 точки. След информацията за родителите в dfs-дървото, за всеки връх, ще запазим колко дървесни ребра "прескача" в унарна бройна система (ако прескача 3 ребра, ще запишем 1110, където 0 ползваме за разделител). Всяка 1 отговаря на едно дървесно ребро, а всяка 0 на един връх, така че това решение ползва $n \log n + 2n$ бита.

Едно решение за 100 точки е да "преместим" погледа си от върховете върху дървесните ребрата. За всеки цикъл, ще маркираме дървесните ребра, които участват в него без най-високото. С други думи едно ребро ще е маркирано, ако то участва в същия цикъл като "родителското" си ребро. Така можем еднозначно да определим какви са били задните ребра в графа.

Тестове и решения: Емил Инджев
Анализ: Иван Лупов