АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
ОПЕРАЦИИ В ДУМИ

R ротации правят пермутацията (*R*+1, *R*+2, …, *N*, 1, 2, 3, …, *R*). Очевидно, ако всички букви в една дума са едни и същи, то тя не се променя нито при реверсия, нито при ротация – такива думи просто преброяваме: те са *T* на брой. Разглеждаме всички реверсии. Нека една от тях е с начало *i* и край *j* ($i \le j$). Това дава пермутацията (1, 2, 3, …, *i*-1, *j*, *j*-1, *j*-2, …, *i*, *j*+1, *j*+2, … *N*). Тъй като двете действия трябва да водят до един и същ резултат, можем просто да разгледаме последната пермутация, ротирана *N*-*R* пъти. Циклите в нея трябва да съответстват на една и съща буква, интересуват ни техният брой и „шаблоните”, които пораждат (можем да си ги мислим като подредени по азбучен ред думи, започващи от `A` и със следващи букви от азбуката, на които, евентуално, са поставени и по-ограничаващи от необходимите условия; всички шаблони се помнят и ако някой вече се е срещал, той и неговите производни не се броят). Нека имаме някакъв „шаблон” с *c* на брой цикъла. Ако буквите във всички цикли са различни, той поражда *T*.(*T*-1).(*T*-2). … .(*T*-c+1) на брой различни думи, като на мястото на „шаблонните” поставим всевъзможните набори от *c* букви в азбуката с *T* букви. Остава да се разгледат възможностите някои от циклите да има едни и същи букви, т. е., да „разбием” множеството от циклите на всевъзможните подмножества от равни елементи. Ако някой от новополучените шаблони вече е срещан (получаван е при предишни разглеждания), той не добавя нови думи към нашето множество.

Съществен момент в този алгоритъм е, че броят на циклите никога не надвишава 9, когато *N* е в дадените граници. Това ни позволява да кодираме всеки шаблон по най-естествения начин – в десетична бройна система с цифрите от 0 до 9, което пък ни дава гаранции, че шаблоните са не повече от 15-цифрени (всъщност, ако кодираме първия символ с 0 – до 14-цифрени) цели числа.

Сложността на този комбинаторен подход се дава от броя на всевъзможните реверсии ($\frac{N(N-1)}{2}+1$), умножени по броя разбивания на множеството от циклите: ако броят на циклите е *c*, това е *c*-тото число на Бел (значи – най-много B_9=`21147`) и по сложността на търсене в структурата от запомнени шаблони, която може да се направи и константна, но и логаритмична е напълно достатъчна. Могат да се правят още оптимизации (най-вече динамични и предварителни пресмятания), задачата може да се изчисли предварително в таблица при тези малки ограничения, но допълнителни „трикове” се оказват ненужни: с този подход алгоритъмът работи достатъчно бързо.

Автор: Павлин Пеев