**АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА**

**ЧИСЛА**

В тази задача състезателите лесно могат да се подведат и да започнат да строят числата от всеки ред. Разбира се, няма нужда да се пазят дълги низове от единици и нули, а двойки числа, които показват броя на единиците и нулите във всяко число от реда. Но дори и след такова съобразяване, задачата никак не е лесна, защото трябва да се пазят двойките числа за последния ред и от тях да се образуват двойките числа (брой на единици и нули) за числата от следващия. Трябва да се съобрязяват размерности на масиви, да се работи с два масива, които алтернативно се сменят, съдържайки последния завършен ред и текущия, който се строи и т.н. – все неща, които са технически трудни за малките състезатели.

Истинското решение предполага друг подход. Първото нещо, което децата трябва да съобразят е, че с изключение на числото 10, което е на първия ред, всички останали числа трябва да съдържат различен брой единици и нули – това следва от начина на образуването им. Това означава, че, ако *K* = *P* и двете числа са по-големи от 1, то число с *K* единици и *P* нули не може да се получи по описания в условието начин. Следователно, за да има шанс да се получи число с *K* единици и *P* нули, трябва *K* ≠ *P*.

Основната идея за решаването на задачата е, че разглеждайки някакво число с *n* единици в началото и *m* нули след това (*n ≠ m*), можем да възстановим числото, от което то се е получило по описания в условието начин, а именно:

Ако *n>m*, то въпросното число се е получило от число с *n-m* единици и *m* нули, намиращо се на предходния ред;

Ако *n<m*, то въпросното число се е получило от число *n* единици и *m-n* нули, намиращо се на предходния ред.

След това наблюдение задачата се решава лесно: започвайки от двойката числа *K* и *P*, трябва да се връщаме назад по редовете, получавайки всеки път числото (като брой единици и нули) от предходния ред, от което би трябвало да се е получило разглежданото число. Ако при това връщане стигнем до числото, което стои на първия ред, т.е. до 10 (една единица и една нула), то началното число с *K* единици и *P* нули може да се получи по описания начин; ако достигнем до число с нула единици или нули, то не може.

Ако обмислим добре описания алгоритъм, то ще видим, че това си е чисто и просто алгоритъмът на Евклид с изваждане за намиране на НОД(*K,P*). Знаем, че той завършва, когато се получат две равни числа, които са равни на търсения НОД. Какво се получи? Ако НОД(*K, P*) = 1, т.е. двете въведени числа са взаимно прости, то число с *K* единици отпред и *P* нули отзад може да се получи по указания в условието начин – в този случай, броят на изважданията плюс 1 ще ни даде на кой ред ще се намира числото. Ако НОД(*K, P*) > 1, то такова число не може да се получи – в такъв случай ще се достигне до стъпка, след която ще се получи равен брой единици и нули, като този брой е по-голям от 0, а такива числа, както отбелязахме в началото не могат да се получат по измисления от Пешо начин.

Както знаем, алгоритъмът на Евклид с изваждане е бавен и идеята, която изложихме по-горе ще донесе 70-80 точки. Веднъж досетили се за алгоритъма на Евклид, можем да приложим техниката му с деление. Тогава много по-бързо ще съумеем да определим на кой ред ще се получи числото (или че няма да може да се получи) и такова решение носи 100 точки. Специфичното тук е, че номерът на реда, на който е числото, ще се получи, като на всяка стъпка от алгоритъма на Евклид добавяме към съответния брояч резултата от делението на по-голямото число на по-малкото.

*Автор: Руско Шиков*