НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА
НАЦИОНАЛЕН КРЪГ 13-15 МАРТ 2026
Група D 6 клас

Анализ на задача Цък

Малки тестове. За всяко число лесно може да се определи колко пъти се казва цък от цифрите му, като то се разглежда цифра по цифра и се гледа колко са петиците и колко са седмиците. За всяко число също може да се намери колко пъти се казва цък от делители, като се дели на 5 докато може и се добавя 1 към броя на всяка стъпка. Аналогично за 7. По този начин може да се намери за всяко число от 1 до **N** колко цък-а биват казани и да се съберат получените резултати.

Сложност – **O(N)**.

Средни тестове. Вече няма да проверяваме всяко число индивидуално, а ще изчислим броя общо. За цифрите ще намерим колко пъти се срещат 5 и 7 на всяка позиция. Нека $N = \overline{a_1 a_2 \dots a_k}$.

Ясно, че 5 и 7 са били цифра на единиците в $P = \overline{a_1 a_2 \dots a_{k-1}}$ числа. Ако $a_k \geq 5$ то 5 се е срещало $P + 1$ пъти. Аналогично ако $a_k \geq 7$ същото важи и за 7. Преминаваме към цифрата на десетиците. Тук става една идея по-сложно. Знаем, че 5 се среща като „петдесетка“ $P = \overline{a_1 a_2 \dots a_{k-2}}$ на брой пъти и съответно $P \times 10^1$ на брой пъти изобщо като цифра на десетиците. Но това не е крайната бройка. Имаме три случая:

1. $a_{k-1} < 5$: тогава нямаме повече „петдесетки“ и сме готови
2. $a_{k-1} = 5$: имаме някаква „непълна петдесетка“ и трябва да добавим към резултата $T = \overline{a_k} + 1$
3. $a_{k-1} > 5$: имаме още една „петдесетка“ която не сме разгледали. Добавяме към резултата 10^1

Разсъжденията за 7 са аналогични.

При стотиците е абсолютно същата логика. Знаем, че 5 се среща като „петстотинтачка“ $P = \overline{a_1 a_2 \dots a_{k-3}}$ на брой пъти и съответно $P \times 10^2$ на брой пъти изобщо като цифра на стотиците. Но това не е крайната бройка. Отново имаме три случая:

1. $a_{k-2} < 5$: тогава нямаме повече „петстотинтачки“ и сме готови
2. $a_{k-2} = 5$: имаме някаква „непълна петстотинтачка“ и трябва да добавим към резултата $T = \overline{a_{k-1} a_k} + 1$
3. $a_{k-2} > 5$: имаме още една „петстотинтачка“ която не сме разгледали. Добавяме към резултата 10^2.

Да разгледаме един пример: N=6735

Позиция	Цифра	Сметка
Единици	5	$673 + 1 = 674$; (+1 защото $a_4 \geq 5$)
	7	$673 + 0 = 673$; ($a_4 < 7$)
Десетици	5	$67 \times 10^1 + 0 = 670$; ($a_3 < 5$)
	7	$67 \times 10^1 + 0 = 670$; ($a_3 < 7$)
Стотици	5	$6 \times 10^2 + 10^2 = 700$; ($a_2 > 5$)
	7	$6 \times 10^2 + 35 + 1 = 636$; ($a_2 = 7$)
Хилядни	5	$0 + 10^3 = 1000$; ($a_1 > 5$)
	7	$0 + 0 = 0$; ($a_2 < 7$)
Общо	$674 + 673 + 670 + 670 + 700 + 636 + 1000 + 0 = 5023$	

За делителите има учудващо елегантно и кратко решение. Може да забележим, че броят на цъковете заради делител 5 са : $\lfloor N/5^1 \rfloor + \lfloor N/5^2 \rfloor + \cdots + \lfloor N/5^K \rfloor$, ($\lfloor x \rfloor$ е най-голямото цяло число по-малко или равно на x) така че 5^K е последната степен на 5 по-малка или равна на **N**. Това е така, защото всяко 5то число се дели на 5, всяко 25-то на 25 и т.н. Затова $\lfloor N/5^i \rfloor$ е броят на числата от 1 до **N**, които се делят на 5^i. Така число, което се дели на 5^i, но не на 5^{i+1} ще се преброи по веднъж от всяка от формулите $\lfloor N/5^1 \rfloor + \lfloor N/5^2 \rfloor + \cdots + \lfloor N/5^i \rfloor$ и ще бъде преброено точно **i** пъти. Аналогично е и решението за 7.

Сложност – **O($\log_{10}$(N)).**

Големи тестове. Остава да модифицираме решението, така че да работи и за числа отвъд long long. Затова ще използваме дълги числа. Нека разгледаме по-подробно какви функции са ни нужни:

1. convert(s) – превръща низ в число, с което ще ни е по-удобно да работим
2. За делителите са нужни две функции:
   a. add(a,b) – събиране на дълго с дълго число
   b. divide(a,b) – делене на дълго с кратко число
3. За цифрите са нужи още две:
   a. mult(a,b) – умножение на дълго с кратко число
   b. mult(a,10^n) – умножение на дълго число със степен на 10ката

Подробности за това как са имплементирани може да намерите в авторовите решения. Понеже извършваме операциите по няколко пъти за всяка цифра на N, крайната сложност е **O($\log_{10}(N)^2$).**

Най-големи тестове. За последните няколко процента се очаква една сравнително сложна, но полезна техника – понеже представяме дългите числа като вектори от int, вместо да пазим цифри от 0 до 9, да се възползваме максимално от пространството, като пазим числа от 0 до 999999999, или иначе казано, да извършваме всички изчисления в бройна система 10^9. Така смаляваме дължината на числата и операциите с тях стават приблизително 9 пъти по-бързи (реално по-малко заради по-голямата константа). Това изисква немалко модификации по кода, а и става доста дълго решение, затова са зададени само последните 10%.

Сложност – **$O(\log_{10}(N)^2 / 9)$.**

Автори: Александър Гатев и Петър Михов