

Weekly Programming Challenge 2

GCD of all permutations of an integer

We know that greatest common divisor of two non-negative integers x, y is the largest number which divides both x and y . We can easily extend this definition to a set of numbers.

Definition : For a set S of non-negative integers, $GCD(S)$ is the largest positive integer which divides each number $x \in S$.

Examples :

For $S = \{12, 30, 84\}$, $GCD(S) = 6$.

For $S = \{12, 30, 121\}$, $GCD(S) = 1$.

Let us consider a specific set $\mathcal{P}(n)$ which is defined for any positive integer n as follows. $\mathcal{P}(n)$ consists of all the numbers which are formed by permuting the digits of n .

Examples :

For $n = 123$, $\mathcal{P}(123) = \{123, 132, 213, 231, 312, 321\}$.

For $n = 8484$, $\mathcal{P}(8484) = \{8484, 8844, 8448, 4488, 4848, 4884\}$.

The challenge is to design an efficient program/algorithm to find the GCD of $\mathcal{P}(n)$ for any given n . Your program should take at most a few seconds (milliseconds if possible) for any arbitrarily large value of n (n may take any positive value valid for `long`).

Note : This challenge does not require any programming tool/technique which we have not covered till now in this course. In other words, the solution of this problem is within reach of all of you.