Reto 2

Carlos Quesada

1. Solución

No existe ningún número con un número par de cifras que cumpla que:

$$n = d_1 d_2 \cdots d_{2k+1} d_{2k} = d_1^{d_2} \cdots d_{2k+1}^{d_{2k}} \tag{1}$$

2. Demostración

Para demostrarlo, realizamos en primer lugar una cota. Sabemos trivialmente que si el número tiene 2k cifras es siempre mayor que 10^{2k-1}. Por otro lado como mucho todas estas cifras pueden ser de valor 9, es decir que el número es como mucho $9^9 + 9^9 + \cdots + 9^9$ k veces, esto es $k \cdot 9^9$. Así que el número tiene que cumplir: $10^{2k-1} < n < k \cdot 9^9$. Pero 10^{2k-1} y $k \cdot 9^9$ son crecientes y la primera diverge mucho más rápido que la segunda. Esto es fácil de comprobar viendo que el límite tiende a infinito cuando k tiende a infinito. Por tanto a partir de un determinado k, 10^{2k-1} será siempre mayor que $k \cdot 9^9$. Comprobamos para k=6 ya se tiene que $10^{11} > 2324522934 = 6 \cdot 9^9$. Así pues, para $k \geq 6$, 10^{2k-1} es siempre mayor que $k \cdot 9^9$, y por tanto no pueden existir soluciones para esos valores de k. Falta por comprobar ahora posibles soluciones para $k6$. Para k=5 tenemos que $k \cdot 9^9$ es igual a 1937102445, por tanto es para los números menores que esta cantidad para los que hay que comprobarlo. En concreto hay que comprobarlo para los números de 2, 4, 6, 8 y 10 cifras que sean menores que 1937102445. Para ello he realizado varios programas cada uno para cada uno de los casos. Podrá haber intentado hacer uno general, también podráa haber intentado un programa algo mas rápido pero Éste, aunque tal vez un poco feo, da unos tiempos más que razonables y no buscaba la apariencia así que vale para la demostración. De hecho inicialmente teníaa un programa que para cada número sacaba las cifras de cada una de sus posiciones pero resultó ser mucho más lento(porque habrá que utilizar módulos y muchas mas operaciones). Los programas funcionan de la siguiente manera: Definimos tantas variables como cifras va a tener nuestro

número. Hacemos un for que recorra todos los números de esa cantidad de cifras. En concreto no empiezo el bucle en 10^{2k-1} sino en $10^{2k-1}+1$ porque de la manera en que trabaja posteriormente empieza a comprobar por $10^{2k-1}+1$. Así nos faltarán por comprobar los número de la forma 10^{2k-1} pero es trivial que estos no cumplen la condición del problema. El interior del bucle consiste sencillamente en ir añadiendo unidades, y si se llega a 10, hacer cero las unidades y sumar una decena, y así sucesivamente para decenas con centenas, centenas y miles etc. Para cada uno de estos valores se comprueba que no se cumple la condición. En la comprobación se exige que no haya ceros en la posición impar para no tener problemas del tipo 0^0. Conviene explicar algo acerca del último, que es algo especial. Como solo hay que calcular hasta 1937102445, sabemos que las dos primeras cifras siempre van a sumar 1 a la cuenta total, así que en el fondo solo hace falta calcular números de 8 cifras, entre 10000000 y 37102445, y la comprobación a realizar es que cada número más 1900000000 ha de ser igual al desarrallo del enunciado de ese número más 1. Con eso se ahorra mucho tiempo de cálculo. Ejecutando los programas se demuestra que no hay ninguna solución para estos casos tampoco.

Programas:

programa 1

```
a=0
b=1
for i in range(11,100):
       a=(a+1)%10
       if a==0:
        b=(b+1)%10
        if b^a==i:
         print(i)
```

```
programa 2
a=0
b=0
c=0
d=1
for i in range(1001,10000):
        a=(a+1)%10
        if a==0:
            b=(b+1)%10
            if b==0:
             c=(c+1)%10
             if c==0:
               d=(d+1)%10
        if b!=0:
          if b^a+d^c==i:
            print(i)

programa 3
a=0
b=0
c=0
d=0
e=0
f=1
for i in range(100001,1000000):
        a=(a+1)%10
        if a==0:
           b=(b+1)%10
           if b==0:
             c=(c+1)%10
             if c==0:
                d=(d+1)%10
                if d==0:
                  e=(e+1)%10
                  if e==0:
                    f=(f+1)%10
        if b!=0 and d!=0:
          if b^a+d^c+f^e==i:
            print(i)
```

```
programa 4
a=0
b=0
c=0
d=0
e=0
f=0
g=0
h=1
for i in range(10000001,100000000):
        a=(a+1)%10
        if a==0:
          b=(b+1)%10
          if b==0:
             c=(c+1)%10
             if c==0:
               d=(d+1)%10
               if d==0:
                  e=(e+1)%10
                  if e==0:
                    f=(f+1)%10
                    if f==0:
                      g=(g+1)%10
                      if g==0:
                        h=(h+1)%10
        if b!=0 and d!=0 and f!=0:
           if b^a+d^c+f^e+h^g==i:
              print(i)
```

```
programa 5
a=0
b=0
c=0
d=0
e=0
f=0
g=0
h=1
for i in range(10000001,37102445):
   a=(a+1)%10
   if a==0:
     b=(b+1)%10
     if b==0:
      c=(c+1)%10
      if c==0:
         d=(d+1)%10
         if d==0:
           e=(e+1)%10
           if e==0:
             f=(f+1)%10
             if f==0:
               g=(g+1)%10
               if g==0:
                 h=(h+1)%10
   if b!=0 and d!=0 and f!=0:
     if b^a+d^c+f^e+h^g+1==i+1900000000:
       print(i)
```