ENUNCIADOS DE LAS ACTIVIDADES DE ÁLGEBRA COMPUTACIONAL

Alejandro Ramírez Rodríguez

22 de diciembre de 2021

- Problema 1: La constante de Champernowne es el número irracional que se obtiene al concatenar todos los números naturales, del siguiente modo:

  $$0.12345678910111213141516171819202122 23...$$

  Se puede ver que el dígito en la posición número 12 después del punto decimal es 1 (indicado en rojo arriba). Sea d_n el dígito que ocupa el lugar n-ésimo. Calcular:

  $$d_1 * d_{10} * d_{10} * d_{1000} * d_{10000} * d_{100000} * d_{1000000}$$

- Problema 2: Se tienen inicialmente 100 cajas (colocadas en círculo) con una bola en cada una. En cada turno, se extraen todas las bolas de una caja y se las coloca, una a una, en las siguientes cajas avanzando en sentido horario. El siguiente turno comienza en la caja donde se colocó la última bola del paso anterior. ¿En qué turno se repite la configuración inicial por primera vez?

- Problema 3: Considerar el siguiente juego de dos jugadores: partiendo de una pila de n piedras, los jugadores van (uno tras otro) quitando 1, 2 o 6 piedras de la pila a su elección, hasta que el que quita la última pierde. Implementar un algoritmo recursivo es posible ganar con n piedras(n) que determine si, partiendo de una pila de n piedras, hay estrategia ganadora.

- Problema 4: Demostrar que es posible multiplicar dos matrices de n×n con $O(n^{\log_2 7})$ operaciones aritméticas. ¿Cuál es la cantidad de operaciones aritméticas si se las multiplica directamente haciendo los productos de filas con columnas? Añadir una gráfica donde se pueda ver la asintótica de la complejidad.

- Problema 5: Un entero $N \geq 2$ se dice pseudoprimo de Fermat en base a, para $a \in Z$ coprimo con N, si $a^{N-1} \equiv 1(mod N)$. Se dice que N es pseudoprimo de Fermat fuerte si es pseudoprimo en cualquier base a coprimo con N. El teorema de Euler-Fermat muestra que todo primo es pseudoprimo de Fermat fuerte, pero lamentablemente la recíproca no es cierta. Los números compuestos $N \geq 2$ que son pseudoprimos de Fermat fuertes se llaman números de Carmichael. Escribir un programa en Python3 que determine los 10 primeros números de Carmichael.

- Problema 6: Utilizar el método $p-1$ de Pollard para factorizar

  $$N = 154220148798056446447985891956740343817921776321968163491478774921 3$$

  utilizando $B = 100$. ¿Cómo se puede calcular $gcd(a^\beta - 1, N)$ de forma eficiente?

- Problema 7: Una versión efectiva del teorema del número primo afirma que

  $$\frac{x}{ln(x)+2} < \pi(x) < \frac{x}{ln(x)-4} \quad \forall x \geq 55$$

donde $\pi(x)$ es la cantidad de números primos en el intervalo $[1, x]$ y $ln(x)$ es el logaritmo natural. En particular, la probabilidad de encontrar un número primo en el intervalo $[1, 10300 - 1]$ eligiendo uniformemente al azar 300 dígitos decimales es aproximadamente $1.45 * 10^{-3}$ uno entre 690. Escribir un programa que genere números de 300 dígitos decimales al azar (dígito a dígito), que aplique el test de Solovay-Strassen con $k = 20$ y que se detenga al encontrar un entero que pase el test, es decir, uno que sea "probablemente primo". ¿Cuántos enteros fueron explorados hasta conseguir el resultado? ¿Qué pasa si se repite el experimento varias veces?

- PROBLEMA 8: Implementar una función que tome un primo $p \neq 2$ y dos polinomios $f, g \in (Z/pZ)[x]$, representados por la lista de sus coeficientes, y que calcule su producto. Para esto, implementar una función recursiva que tome $[f], [g] \in (Z/pZ)[x]/ < x^{2^k} + 1 >$ y calcule su producto aplicando el método de Schönhage-Strassen, cuyo coste es de $O(n * \log(n) * \log(\log(n)))$