Power of Cryptography
Johanna Beltran y Diego Triviño
2012

Índice

1. Introducción

'**Power of Cryptography**' es un problema de programación el cual encontramos en el juez virtual UVA con el número **113**.
Este documento busca mostrar una de las tantas soluciones desde el enfoque matemático teniendo en cuenta que el objetivo es realizar la implementación de la solución del problema en cualquier lenguaje de programación con la ayuda de este documento.
Este problema puede ser resuelto utilizando la estrategia de 'divide y vencerás'.
Esta estrategia es una técnica de diseño de algoritmos la cual consiste en dividir de forma recurrente un problema en subproblemas más sencillos hasta que se encuentre un caso base.

Esta técnica consta fundamentalmente de los siguientes pasos:

1. Descomponer el problema a resolver en un cierto número de subproblemas más pequeños.
2. Resolver independientemente cada subproblema.
3. Combinar los resultados obtenidos para construir la solución del problema original.

2. Definición del problema

Este problema implica el cálculo eficiente de la raíz entera de un conjunto de números.

Se deben tener en cuenta las siguientes restricciones para la solución del problema:

1. Para cada caso de prueba se debe ingresar un número $n \geq 1$ y un número $p \geq 1$
2. Para todos los pares de números ingresados se tiene en cuenta que $1 \leq n \leq 200$ y $1 \leq p \leq 10^{101}$.

2.1. Entrada

La entrada consiste en una secuencia de pares de números enteros n y p con cada número entero en una línea diferente. Para todos los pares $1 \leq n \leq 200$, $1 \leq p \leq 10^{101}$ existe un entero k, $1 \leq k \leq 10^9$ tal que $k^n = p$.

EJEMPLO

2

16

3

27

7

4357186184021382204544

2.2. Salida

Para cada par de enteros n y p el valor $\sqrt[n]{p}$ debe ser impreso, es decir, el número k de tal manera que $k^n = p$.

EJEMPLO ANTERIOR

4

3

1234

3. Modelamiento matemático

Dados dos números enteros positivos n y p se debe plantear un k tal que $k^n = p$.

4. Planteamiento de la solución

Para cada caso de prueba se tendrá un número k que varía entre $1 \leq k \leq (p \div n)$ hasta que $k^n = p$ en el cual se utilizará una función recursiva para calcular los exponentes de un número.

La función recursiva para calcular los exponentes de un número será la siguiente:

$$expo(p,n) = \begin{cases} 1, & \text{si n= 0} \\ \text{expo(k,n/2)} \times \text{expo } (k,n/2)\ , & \text{si n es par} \\ \text{expo k} \times (k, n-1)\ , & \text{si n es impar} \end{cases}$$

5. Conclusiones

1. Por las características de este problema se recomienda utilizar el enfoque de la estrategia de 'divide y vencerás' puesto que cada problema se reduce a un único subproblema más simple que un algoritmo general.
2. Para este problema utilizamos una función recursiva para calcular los exponentes de un número que torna el problema a ser más eficiente, sin embargo, se pueden implementar algoritmos no recursivos que almacenen las soluciones parciales en una estructura de datos explícita, como puede ser una pila o cola.