



Universidad
Carlos III de Madrid

Grado en Ingeniería Informática

Curso 2020/2021

Teoría Avanzada de la Computación

Test de Primalidad - AKS

Hito 2

Autores:

Iván Miguelez García	100383387
Alba Reinders Sánchez	100383444
Alejandro Valverde Mahou	100383383

Índice

1. Hito 2: Estudio de la complejidad del <i>Totient</i>	3
1.1. Estudio analítico	3
1.2. Estudio empírico	3

1. Hito 2: Estudio de la complejidad del *Totient*

En esta segunda parte se pide realizar el estudio analítico y empírico del cálculo del *Totient* ($\phi(r)$). Donde $\phi(r)$ es el número de enteros positivos más pequeños o iguales que r tales que r es coprimo con ellos, es decir, su *med* es 1.

1.1. Estudio analítico

A continuación, se realiza el estudio analítico para averiguar la complejidad temporal del algoritmo. Analizando el código se ve que la función del cálculo del *Totient* está formada por un bucle *for* externo y un bucle *while* interno:

El bucle de fuera se ejecuta como mucho $\lfloor \sqrt{n} \rfloor$ veces, dado que en este caso, la peor situación se da cuando n es un número primo, y por tanto el bucle de fuera tiene que recorrer desde $i = 2$ hasta $i = \lfloor \sqrt{n} \rfloor + 1$. Simplificando, se encuentra \sqrt{n} veces.

El bucle de dentro se ejecuta como mucho $\log_i n$ veces, porque el peor caso resulta cuando $n = i^k$ donde k es un número entero. Por tanto, despejando, $k = \log_i n$, y k representa el número de veces que se realiza el bucle.

Para que se cumpla el peor de los casos del bucle *for* exterior, n tiene que ser un número primo, y por tanto el bucle *while* interior no se realizará ninguna vez. En el caso de que n sea el número primo, también se obtiene el valor máximo de $\phi(n)$, que es $n - 1$.

1.2. Estudio empírico