Ciencia de la computación

Laboratorio 02

Álgebra Abstracta

Juan Pablo Galindo Coronel

Tercer semestre

2021

"El alumno declara haber realizado el presente trabajo de acuerdo a las normas de la Universidad Católica San Pablo"

FIRMA

1. (5 points) Calcular el tiempo computacional del Algoritmo de Euclides. Detalle y sustente su respuesta.

Algoritmo de Euclides es un método eficiente para encontrar el MCD (mayor divisor común) de dos números enteros. La complejidad temporal de este algoritmo es
O (log (min (a, b))
Supongamos que ayb son dos números enteros tales que a> b entonces de acuerdo con el algoritmo de Euclides:

mcd (a, b) = mcd (b, a% b)

Utilice la fórmula donde b es 0 . En este paso, el resultado será el MCD de los dos enteros , que será igual a a . Entonces, después de observar cuidadosamente, se puede decir que la complejidad temporal de este algoritmo sería proporcional al número de pasos necesarios para reducir b a 0 .Asumamos que el número de pasos necesarios para reducir b a 0 usando este algoritmo es N

mcd (a, b)= N veces

Ahora,si el euclidiana Algoritmo para dos números a y b se reduce en N pasos entonces, una debe ser al menos f (N + 2) y b debe ser al menos f (N + 1).

mcd (a, b) ——> N pasos
Entonces, a> = f (N + 2) yb> = f (N + 1)
donde, f N es el (N- ésimo) término en la serie de Fibonacci (0, 1, 1, 2, 3,…)
y N > = 0.
Para probar la afirmación anterior utilizando el principio de inducción matemática (PMI)
Caso Base

- Supongamos que a = 2 y b = 1 . Entonces, mcd (2, 1) se reducirá a mcd (1, 0) en 1 paso, es decir, N = 1 .
- Esto significa que 2 debe ser al menos f 3 y 1 debe ser al menos f 2 y f 3 = 2 y f 2 = 1 .
- Esto implica que a es al menos f (N + 2) y b es al menos f (N + 1) .
- Se puede concluir que la afirmación es cierta para el caso base.
- Paso inductivo: Supongamos que la afirmación es válida para la (N - 1) º Paso . Entonces, a continuación se muestran los pasos para
  demostrar N- ésimo paso
  mcd (b, a% b) ——> (N - 1) pasos
  Entonces,

  b> = f (N - 1 + 2) es decir, b> = f (N + 1)
  a% b> = f (N - 1 + 1) es decir, a% b> = f

- Ahora, (a / b) siempre sería mayor que 1. Entonces, del resultado anterior, se concluye que
  a> = b + (a% b)
  Esto implica, a> = f (N + 1) + f N
  se puede decir que si el algoritmo de Euclides para dos números a y b se reduce en N pasos entonces, una debe ser al menos f (N + 1)

Tenemos que tenemos la formula bliter

f N = {((1 + √5) / 2) N - ((1 - √5) / 2) N } / √5
o
f N ≈ ∅ N

donde, ∅ se conoce como la proporción áurea (∅≈1.618) , y f N es el número N de Fibonacci .

Ahora, ya se ha dicho que la complejidad del tiempo será proporcional a N, es decir, el número de pasos necesarios para reducir b a 0 .

Entonces, para probar la complejidad del tiempo, se sabe que:

f N ≈ ∅ N

N ≈ log ∅ (f N)

Ahora, a partir de la declaración anterior, se demuestra que el uso del principio de inducción matemática, se puede decir que si el algoritmo de Euclides para dos números a y b se reduce en N pasos entonces, una debe ser al menos f (N + 2) y b debe ser al menos f (N + 1) .

De los dos resultados anteriores, se puede concluir que:

=> f N + 1 ≈ min (a, b)

=> N + 1 ≈ log ∅ min (a, b)

=> O (N) = O (N + 1) = log (min (a, b))

2. (15 points) Implementar el Algoritmo Extendido de Euclides. Implementar un programa que permita ingresar dos números a y b (enteros positivos) y que retorne {gcd(a, b), x, y}, donde gcd(a, b) = ax + by (x, y ∈ Z). Enviar un enlace al repositorio (´única forma de envío). En este colocara un README.md con una breve descripción del programa y las instrucciones para ejecutarlo. Colocar un ejemplo del resultado esperado.

   EN esta ocasión mi compilador fue :

https://github.com/juanpa022/Extendido-de-Euclides

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
Primer valor: 486
Segundo valor: 1260
 X=> 13
 Y=> -5
MCD = (  486 ,  1260 ) == 18
>
```