Problema A

Ricardo Enrique González

Camilo Zambrano

201425733

201515438

Como podemos observar nos hace falta la complejidad temporal de encontrar el MCD, para la complejidad temporal de este algoritmo existen diversos criterios, algunos de ellos muy diferentes, sin embargo en general la mayoría de ellos concuerdan en que la complejidad es logarítmica, lo cual tiene sentido debido a que cada vez que intentamos simplificar estamos dividiendo, es decir, estamos disminuyendo logarítmicamente la cantidad de n, entre ellos está el teorema Lames que se encuentra como una de nuestras referencias en {5] y {6] y entre otras y la que decidimos usar es de un artículo que se encuentra en nuestras referencias [4] en el cual se menciona que: ‘he computational efficiency of Euclid's algorithm has been studied thoroughly. This efficiency can be described by the number of division steps the algorithm requires, multiplied by the computational expense of each step. The first known analysis of Euclid's algorithm is due to A.-A.-L. Reynaud in 1811,mwho showed that the number of division steps on input (*u*, *v*) is bounded by *v*; later he improved this to *v*/2 + 2. Later, in 1841, P.-J.-E. Finck showe that the number of division steps is at most 2 log2 *v* + 1’. Por tanto esta será la complejidad que usaremos puesto que es bastante práctica de usar.

T(n) = O(Max( (m - #trans), (n - #per), Log(v)) )

S(n) = θ (13) = θ (1)

3. *Comentarios finales*

El algoritmo cumple eficazmente con todos los casos de prueba presentados por el enunciado y otros casos de prueba más estrictos propuestos por el grupo de trabajo. Adicionalmente el tiempo de respuesta observado por este parece ser satisfactorio con respecto a los estándares del curso y no se encontraron soluciones más óptimas en complejidad, por tanto consideramos que el algoritmo propuesto es el adecuado para la solución del problema.

Nota: Al correr el archivo si no imprime el ultimo valor de inmediato, darle 2 veces a enter.

Referencias:

[1] <https://en.wikipedia.org/wiki/Euclidean_algorithm>

[2] <https://cgi.csc.liv.ac.uk/~martin/teaching/comp202/Java/GCD.html>