|  |
| --- |
| ESCUELA SUPERIOR DE INFORMÁTICA, UCLM  15/03/2021 |

|  |
| --- |
| HITO 1 |
| PRACTICA 2 AMONG US |
| **Algoritmos Divide y vencerás** |

**ALUMNOS:**

Esther Camacho Caro

Diego Dorado Galán



Contenido

[Tareas a realizar: 2](#_Toc66742854)

[Complejidad teórica Algoritmo Divide y vencerás 3](#_Toc66742855)

[posicionImpostor 3](#_Toc66742856)

# Tareas a realizar:

1. Implementar un programa Java que descubra al impostor con el mínimo uso del medidor de ira posible, para prevenir que éste se queme. El programa deberá mostrar la siguiente información:

* Número de jugadores (se habrá leído por teclado previamente).
* Quién es el impostor. Se podrá mostrar información como su número de tareas completadas, experiencia, posición en la lista, etc. También se podrá mostrar información adicional sobre el proceso de búsqueda del impostor: pasos que se van dando en la búsqueda del impostor, los usos que se van realizando del medidor de ira, etc.
* Quién es el ganador (el impostor o la tripulación). Se podrá mostrar información como su número de tareas completadas, experiencia, posición en la lista, etc.
* Tiempo (en nanosegundos) que se ha tardado en encontrar al impostor.Determinar la complejidad teórica de cada algoritmo implementado.

1. Ejecutar el juego varias veces con varios números de jugadores (entre 3 y N). Comprobar empíricamente la cantidad de jugadores aproximada que soporta vuestro PC.
2. Proporcionar un documento en el que se detalle lo siguiente:

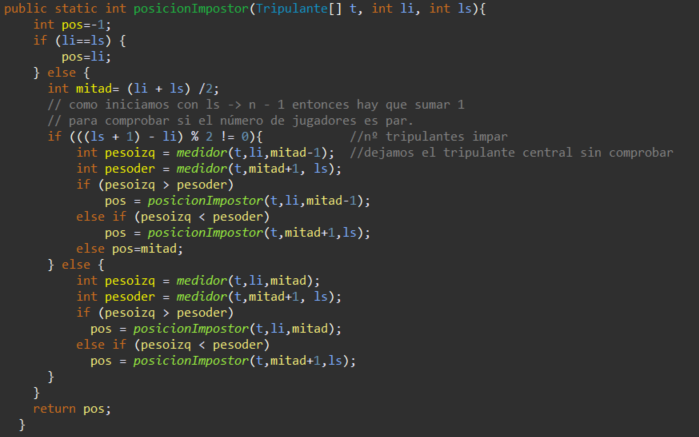
a. Explicación breve de la estrategia seguida para descubrir al impostor.

b. Determinar la complejidad teórica del algoritmo principal que habéis implementado (búsqueda del impostor).

c. Explicar los resultados de la comprobación empírica realizada en el punto 2 anterior sobre el número de jugadores.

# Complejidad teórica Algoritmo Divide y vencerás

## posicionImpostor



T(n) = T(n/2) + 1

n=2m → m=log(n)

t(2m) = t(2m/2) +1

tm – tm-1 = 1

bm \* p(m)d

1m \* m0

xm – xm-1 = 1

xm-1 (x – 1) = 1

(x-1)\*(x-1)

tm = C1 (1)m + C2 m(1)m  → tm = C2 m(1)m  **O(logn) Complejidad logarítmica**

**ESTRATEGIA EMPLEADA**

Hemos empleado un algoritmo divide y vencerás que se basa en la búsqueda binaria. El algoritmo desarrollado es muy similar al que se nos ha facilitado en las clases de teoría de la propia asignatura. Distinguimos tres momentos: el primero de ellos, cuando el límite superior y el inferior coinciden, que implicaría que el impostor estaría en esa posición; número de jugador par, donde debemos analizar únicamente cual de las dos mitades tiene un valor de ira mayor (ya que nos indicará en que fracción está el impostor) y realizar una llamada recursiva hasta acotar a una única posición al impostor; y por último, si el número de jugadores es impar, dónde sumado a la estrategia del caso de jugadores pares, deberemos distinguir una posición central restante, dónde se situará el impostor en caso de que las dos mitades (con el mismo número de jugadores, de ahí la distinción de la posición central) sean de igual valor.

**CONCLUSIÓN**

Podemos extraer de esta práctica que un algoritmo divide y vencerás, o al menos el empleado en nuestra práctica, aumenta en tiempo de ejecución al aumentar el valor del rango de jugadores o de valores sobre los cuales acotar u operar. Basándonos empíricamente en las pruebas realizadas en nuestros computadores, soportan prácticamente hasta el límite de valores de un número entero, a partir del cual saltará una excepción por superar los valores permitidos para este tipo de variable (2 147 483 648), pero no consideramos que sea una práctica adecuada debido al excesivo tiempo de ejecución que conlleva. Además, en nuestro caso personal, imprimimos el listado completo de jugadores creados (junto a sus atributos), lo que aumenta el tiempo de ejecución, limitado por la impresión por pantalla de dichos datos.