

## Laboratorio Nro. 2

### Fuerza Bruta (Brute force o Exhaustive search)

**Simón E. Correa Henao**  
Universidad Eafit  
Medellín, Colombia  
scorreah@eafit.edu.co

**David Gómez Correa**  
Universidad Eafit  
Medellín, Colombia  
dgomezc10@eafit.edu.co

#### 3) Simulacro de preguntas de sustentación de Proyectos

**3.1** El código para conocer el camino más corto para recorrer todos los vértices de un grafo, consta de un algoritmo que evalúa por fuerza bruta todas las posibles maneras de recorrer el grafo y conocer el costo, para posterior mente evaluar camino por camino y tomar el más corto. Se plantea una condición de parada qué es cuando se recorre todos los vértices y el ultimo vértice corresponde al vértice de partida.

**3.2** La complejidad asintomática para el peor de los casos seria de  $O(V \cdot E)$  puesto que el algoritmo recorre todos los vértices ( $V$ ) del grafo, para posterior mente evaluar cada una de sus Aristas( $E$ ) pues al ser fuerza bruta va a ser necesario preguntar por cada uno de los caminos.

**3.3** El algoritmo para solucionar ese problema en específico sería posible, el tiempo de ejecución dependería de la cantidad de Aristas presentes en cada uno de los vertices en este caso clientes, suponiendo que cada uno de los clientes está conectado con cualquier otro cliente, el tiempo total de ejecución seria de 2500 milisegundos, es decir alrededor de 2.5 segundos, esto claro dependiendo de la velocidad del procesador y suponiendo que se demora un milisegundo verificando un cliente.

**3.4** Para darle solución al problema utilizamos dos estructuras de datos, una matriz que almacena el tablero lleno de puntos (.) y o asteriscos (\*) cuando se trata de una casilla bloqueada. El algoritmo funciona igual que el de  $n$  reinas normal, evalúa si lo puedo colocar en una posición de tal manera que la pendiente con otra posición no sea igual, y que no estén en la misma fila, el único cambio adicional es no utilizar las casillas bloqueadas, es decir no utilizarlas como una posible solución.

**3.5** La complejidad para el peor de los casos en este ejercicio es de  $O(n^2)$  puesto que es necesario recorrer todas las reinas es decir el ancho del tablero y después verificar si se puede colocar en alguna fila de cierta columna que sería  $n$ ; por tal motivo el algoritmo tiene dicha complejidad.

## ESTRUCTURA DE DATOS 2

### Código ST0247

**3.6** La única variable que utilizamos es  $n$ , y su significado se ve reflejado en el tamaño del tablero  $n \times n$  y la cantidad de reinas a colocar, es decir las  $n$  reinas.

#### 4) Simulacro de Parcial

**4.1** 1.  $actual > maximo$   
2.  $O(n^2)$

**4.2** 1.  $i-j$   
2.  $n$   
3.  $O(n)$

**4.3** No Lo se

**4.4** 1.  $i + 1$   
2.  $left == right$

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473

