

Laboratorio Nro. 3

Fuerza Bruta (*Brute force o Exhaustive search*)

Objetivos

- ☒ Diseñar algoritmos usando la técnica de diseño de búsqueda por fuerza bruta

Instrucciones Generales

Leer la Guía del Laboratorio



Antes de comenzar a resolver el presente laboratorio, leer la “**Guía Metodológica para la realización y entrega de laboratorios de Estructura de Datos y Algoritmos**” que les orientará sobre los requisitos de entrega para este y todos los laboratorios, las rúbricas de calificación, el desarrollo de procedimientos, entre otros aspectos importantes.

Completar Autoevaluación



Como **actividad valorativa**, deben diligenciar el formato de autoevaluación a través del siguiente enlace: <http://bit.ly/2g8T8O6> . Haciendo esta evaluación es la única forma de alcanzar 5.0

Registrar Reclamos



En caso de tener **algún comentario** sobre la nota recibida en este u otro laboratorio, pueden **enviarlo** a través de <http://bit.ly/2g4TTKf>, el cual será atendido en la menor brevedad posible.

Entregar Archivos



Los archivos que **ustedes deben entregar** al docente son: **un archivo PDF** con el informe de laboratorio usando la plantilla definida, y **dos códigos**, uno con la solución al numeral 1 y otro al numeral 2 del presente. Estos últimos **comprimidos** en un único ZIP.

Traducción de Enunciados



En el ZIP que el docente entrega a los alumnos, encontrarán la traducción al español de los enunciados de los Ejercicios en Línea.

Visualización de Calificaciones



A través de **Eafit Interactiva** encontrarán **un enlace** que les permitirá **ver un registro de las calificaciones** que **emite el docente** para cada taller de laboratorio y según las rubricas expuestas. **Véase Guía en sección 3, numeral 3.8.**

Resolver Ejercicios

1. Códigos para entregar en un ZIP junto la documentación en HTML:



En la vida real, la documentación del software hace parte de muchos estándares de calidad como CMMI e ISO/IEC 9126



Véase Guía **en Sección 3, numeral 3.4**



Código de laboratorio en **ZIP**



Documentación en **HTML**



No se reciben archivos en **.RAR**



En la vida real, la resolución de acertijos como las *N reinas* es una pregunta frecuente en entrevistas técnicas de grandes como compañías como Google, Microsoft y Facebook. Léase más en <http://bit.ly/2hH0xIx>

1.1 Teniendo en cuenta lo anterior Implementen el algoritmo de fuerza bruta para encontrar **TODAS** las soluciones de las N Reinas.

1.2 Construyan ejemplos usando JUnit para probar su implementación de las N Reinas usando fuerza bruta.

Como muestra, use los ejemplos trabajados en el taller de clase sobre fuerza bruta para el problema de las 4 reinas



NOTA: Si utilizan Python o C++, utilice una librería equivalente para pruebas unitarias en dichos lenguajes.



Pista: Véase Guía, **Sección 4, numeral 4.14** “Cómo hacer pruebas unitarias en BlueJ usando JUnit”



NOTA: Todos los ejercicios del numeral 1 deben ser documentados en formato HTML. Véase Guía en **Sección 4, numeral 4.1** “Cómo escribir la documentación HTML de un código usando JavaDoc”

2) Ejercicios en línea sin documentación HTML en el ZIP



Véase Guía en **Sección 3, numeral 3.3**



Entregar un archivo en **.JAVA**



O entregar un archivo en **.CPP**



No entregar documentación **HTML**



O entregar un archivo en **.PY**



No se reciben archivos en **.PDF**



NOTA: Recuerden que, si toman la respuesta de alguna fuente, deben referenciar según el tipo de cita correspondiente. Véase *Guía en Sección 4, numerales 4.16 y 4.17*

2.1 Resuelvan el siguiente problema: <http://bit.ly/2n8b1iy>



PISTA: Una solución de fuerza bruta no les permitiría pasar el tiempo máximo que pide el juez. No es obligatorio que sea aprobado por el juez, pero es para tenerlo en cuenta. Lo mejor es usar *Backtracking*.



PISTA: Primero defina una estructura de datos para representar los huecos



PISTA 2: Véase Guía en **Sección 4, numeral 4.13** “Cómo usar Scanner o *BufferedReader*”

3 Preguntas para resolver en el informe PDF



Véase Guía en **Sección 3**,
Numeral 3.5



Entregar informe de
laboratorio en **PDF**



Usen la **plantilla** para
responder laboratorios



**No apliquen Normas
Icontec** para esto



En la vida real, las técnicas usadas para resolver muchos
problemas en Ingeniería de Sistemas son las mismas que las
existentes para las N reinas

3.1 Teniendo en cuenta lo anterior, respondan: Para resolver el problema de las N Reinas, fuera de fuerza bruta, ¿qué otras técnicas computacionales existen?



PISTA : Léase <http://bit.ly/2hvFCb2>

3.2 Tomen los tiempos de ejecución del programa implementado en el numeral 1.1 y completen la siguiente tabla. Si se demora más de 5 minutos, coloque “se demora más de 50 minutos”, no sigan esperando, podría tomar siglos en dar la respuesta, literalmente.

Valor de N	Tiempo de ejecución
4	
8	
16	

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

32	
N	$O(?)$



PISTA: Véase *Guía en Sección 4, numeral 4.6* “Cómo usar la escala logarítmica en Microsoft Excel 2013”

3.3 Escriban una explicación entre 3 y 6 líneas de texto del código del ejercicio en línea del numeral 2.1. Digan cómo funciona, cómo está implementado y destaquen las estructuras de datos y algoritmos usados

3.4 Expliquen con sus propias palabras la estructura de datos que utiliza para resolver el problema del numeral 2.1 y cómo funciona el algoritmo.



NOTA: Recuerden que debe explicar su implementación en el informe PDF

3.5 Calculen la complejidad del ejercicio en línea del numeral 2.1 y agréguela al informe PDF



PISTA: Véase *Guía en Sección 4, numeral 4.11* “Cómo escribir la complejidad de un ejercicio en línea”

3.6 Expliquen con sus palabras las variables (qué es ‘n’, qué es ‘m’, etc.) del cálculo de complejidad del numeral anterior

4) Quiz de concepto teóricos tipo Saber Pro para resolver en el informe PDF



PISTA: Véase *Guía en Sección 4, Numeral 4.18* “Respuestas del Quiz”



PISTA 2: Lean las diapositivas tituladas “*Data Structures II: Brute Force*”, encontrarán la mayoría de las respuestas

1. La estrategia de fuerza bruta para resolver un problema computacional consiste en:

- a) Dividir el problema en dos subproblemas sucesivamente
- b) Enumerar todas las posibles soluciones y seleccionar una o varias
- c) Buscar aleatoriamente algunas soluciones y seleccionar la mejor entre ellas
- d) Escoger vorazmente la mejor solución

2. Para resolver las n-reinas por fuerza bruta es necesario:

- a) Dividir el problema de n reinas en dos problemas de $n/2$ reinas
- b) Enumerar las $n!$ Permutaciones de reinas y mirar en cuáles no se atacan las reinas
- c) Buscar aleatoriamente algunas soluciones y seleccionar la mejor entre ellas
- d) Escoger vorazmente la mejor solución

3. ¿Cómo se representa más eficientemente el tablero de una solución de las N reinas?

- a) Con una cadena de caracteres de tamaño $N \times N$
- b) Un arreglo de tamaño N
- c) Una matriz de tamaño $N \times N$
- d) Una lista de listas de tamaño $N \times N$

4. ¿Cuál es la complejidad de resolver el problema del agente viajero por fuerza bruta?

- a) $O(n)$
- b) $O(n^2)$
- c) $O(\log n)$
- d) $O(n!)$

DOCENTE MAURICIO TORO BERMÚDEZ

Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627

Correo: mtorobe@eafit.edu.co

5. ¿En qué consiste el problema de las N Reinas?

- a) Ubicar N-1 reinas en un tablero de NxN de ajedrez de tal forma que no se ataquen las reinas
- b) Ubicar N reinas en un tablero de NxN de ajedrez de tal forma que no se ataquen las reinas
- c) Ubicar N+1 reinas en un tablero de NxN de ajedrez de tal forma que no se ataquen las reinas
- d) Ubicar N^2 reinas en un tablero de NxN de ajedrez de tal forma que no se ataquen las reinas

5. [Ejercicio Opcional] Lectura recomendada

"Quienes se preparan para el ejercicio de una profesión requieren la adquisición de competencias que necesariamente se sustentan en procesos comunicativos. Así cuando se entrevista a un ingeniero recién egresado para un empleo, una buena parte de sus posibilidades radica en su capacidad de comunicación; pero se ha observado que esta es una de sus principales debilidades..."

Tomado de <http://bit.ly/2gJKzJD>



Véase Guía en **Sección 3, numeral 3.6 y 4.20** de la Guía Metodológica, "Lectura recomendada" y "Ejemplo para realización de actividades de las Lecturas Recomendadas", respectivamente

Posterior a la lectura del texto *Anany Levitin, Introduction to the Design & Analysis of Algorithms Chapter 3: Brute Force and Exhaustive Search. Páginas*

97 – 120.”, realicen las siguientes actividades que les permitirán sumar puntos adicionales:

- a) Escriban un resumen de la lectura que tenga una longitud de 100 a 150 palabras



PISTA: En el siguiente enlace, unos consejos de cómo hacer un buen resumen <http://bit.ly/2knU3Pv>



PISTA 2: [Aquí](#) le explican cómo contar el número de palabras en Microsoft Word

- b) Hagan un mapa conceptual que destaque los principales elementos teóricos.



PISTA: Para que hagan el mapa conceptual se recomiendan herramientas como las que encuentran en <https://cacoo.com/> o <https://www.mindmup.com/#m:new-a-1437527273469>



NOTA: Si desean una lectura adicional en inglés, consideren la siguiente: “Skiena, *The algorithm design manual* (2nd edition), Section 14.4. 2010”, que pueden encontrarla en biblioteca.



NOTA 2: Estas respuestas también deben incluirlas en el informe PDF

6. [Ejercicio Opcional] Trabajo en Equipo y Progreso Gradual



El trabajo en equipo es una exigencia actual del mercado. "Mientras algunos medios retratan la programación como un trabajo solitario, la realidad es que requiere de mucha comunicación y trabajo con otros. Si trabajas para una compañía, serás parte de un equipo de desarrollo y esperarán que te comuniques y trabajes bien con otras personas"

Tomado de <http://bit.ly/1B6hUDp>



Véase Guía en **Sección 3, numeral 3.7** y **Sección 4, numerales 4.21, 4.22 y 4.23** de la Guía Metodológica

- a) Entreguen copia de todas las actas de reunión usando el tablero Kanban, con fecha, hora e integrantes que participaron



PISTA: Véase **Guía en Sección 4, Numeral 4.21** “Ejemplo de cómo hacer actas de trabajo en equipo usando Tablero Kanban”

- b) Entreguen el reporte de *git*, *svn* o *mercurial* con los cambios en el código y quién hizo cada cambio, con fecha, hora e integrantes que participaron



PISTA: Véase Guía en Sección 4, Numeral 4.23 “Cómo generar el historial de cambios en el código de un repositorio que está en svn”

- c) Entreguen el reporte de cambios del informe de laboratorio que se genera *Google docs* o herramientas similares



PISTA: Véase Guía en Sección 4, Numeral 4.22 “Cómo ver el historial de revisión de un archivo en Google Docs”



NOTA: Estas respuestas también deben incluirlas en el informe PDF