

Proyecto final Análisis y diseño de algoritmos

Tipo:

Proyecto Final

Objetivo:

Resolver el algoritmo bajo una técnica específica y calcular la complejidad algorítmica usando la notación Big O


Condiciones de Entrega

Condición	Descripción
Fecha de entrega	27 de mayo de 2020
Modo de entrega	Código fuente y documentación
Entrega	Individual
Rubrica evaluación	Sustentación en clase

DEFINICION DEL PROYECTO:

- Resolver el algoritmo según la técnica establecida para cada uno.
- Usar lenguaje de programación Python .

NOMBRE	TECNICA	ALGORITMO
ALEJANDRO PADILLA GUEVARA	Divide y vencerás	karatsuba
ARIEL MEJÍA SUAREZ	Divide y vencerás	búsqueda binaria
BRIAN LEONARDO RIAÑO SANTANA	Divide y vencerás	Búsqueda de raíces
CAMILO ANDRES DIAZ GOMEZ	Programación dinámica	Floyd
JEFERSON ANDREY DAZA CARO	Algoritmos voraces	Kruskal
JIMMY ALEJANDRO RIOS PIÑEROS	Algoritmos voraces	Prim
JUAN DANIEL AVILA CAICEDO	Backtracking	Las ocho reinas
JUAN DAVID GONZALEZ DIMATÉ	Programación dinámica	WARSHALL
JUAN ESTEBAN CONTRERAS DIAZ	Backtracking	Problema del caballo ajedrez
JUAN NICOLAS MONTES CETINA	Backtracking	Sudoku
JUAN SEB RODRIGUEZ CORREA	Ramificación y Poda	Problema del viajante
LUIS FELIPE VELASCO TAO	Programación dinámica	el problema de la mochila
MANUEL EDUARDO ALBARRAN PEREZ	Algoritmos voraces	Dijkstra
NICOLAS CARO ÑUSTEZ	Ramificación y Poda	Recorridos árbol B*
SANTIAGO ECHEVERRI DUQUE	Ramificación y Poda	Recorridos árbol rojo negro
SERGIO ALEJANDRO LÓPEZ DELGADO	Algoritmos voraces	Bellman - Ford

 UNIVERSIDAD DE SAN BUENAVENTURA SEDE BOGOTÁ	Facultad de Ingeniería Ingeniería de Sistemas	Análisis de algoritmos
---	--	------------------------

1. Actividades

1. Explicar en qué consiste la técnica de programación. (Definición, características, uso)
2. Explicar y dar un contexto en qué consiste el problema del algoritmo.
3. Explicar como va a resolver el algoritmo. (descripción del algoritmo, características, elementos clave)
4. Programar el algoritmo en lenguaje Python (debe tener interacción con el usuario para leer o escribir los datos necesarios para ejecutar el algoritmo, ya sea por consola o interfaz gráfica) (Nota: No se requiere conexión a base de datos o archivos planos)
5. Para cada algoritmo usando en resolver el problema calcular
 - Calcular la complejidad algorítmica en el mejor de los casos
 - Calcular la complejidad algorítmica en el caso promedio
 - Calcular la complejidad algorítmica en el peor de los casos (Usar Notación Big O)
6. Documentar el código fuente

2. Entrega

- Enviar vía corre electrónico
- En un solo archivo comprimido (ZIP, RAR, TAR, JAR o GZIP)
- Reporte (DOC, DOCX o PDF)
- Códigos fuente (.py.)
- Sustentación