

PROGRAMA DE CURSO

| Código | Nombre | | | |
|-----------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|------------------|-------------------------|---------------------------|
| | Trabajo Dirigido: Análisis Adaptiva de la Computación de Códigos Prefijos Óptimos, y variantes | | | |
| Nombre en Inglés | | | | |
| Research Assistantship: Adaptive Analysis of the computation of Optimal Prefix Free Codes and Variants | | | | |
| SCT | Unidades Docentes | Horas de Cátedra | Horas Docencia Auxiliar | Horas de Trabajo Personal |
| 3 | 10 | 0 | 0 | 10 |
| Requisitos | | | Carácter del Curso | |
| CC3001 | | | Electivo | |
| Resultados de Aprendizaje | | | | |
| El alumno que ha aprobado este curso : | | | | |
| 1. habrá profundizado su entendimiento de los temas vistos en CC4102 tales que | | | | |
| 1) "Greedy Algorithms", | | | | |
| 2) "Optimal Prefix Free Codes", | | | | |
| 3) "Computational Lower Bounds in the model of comparisons", y | | | | |
| 4) "Dictionary Data Structures"; | | | | |
| 2. habrá aprendido técnicas blandas de investigación en Informática Teórica: | | | | |
| 1) Como leer y estudiar un articulo de investigación, | | | | |
| 2) Como realizar un estudio teórico, y | | | | |
| 3) Como redactar científicamente los resultados encontrados; | | | | |
| 3. habrá aprendido técnicas específicas de investigación sobre el tema del análisis adaptiva de algoritmos y estructuras de datos | | | | |
| 1) Como definir una medida de dificultad, | | | | |
| 2) Como diseñar un nuevo algoritmo, y | | | | |
| 3) Como mostrar una cota inferior sobre la complejidad computacional; | | | | |
| 4. habrá aprendido varios temas específicos de informática teórica | | | | |
| 1) "Alphabetic Search Trees" | | | | |
| 2) "Minimax Trees" | | | | |

| Metodología Docente | Evaluación General |
|-----------------------|-----------------------------------------------------------------------------|
| - Reuniones semanales | - Evaluaciones mensuales - Redaccion de un reporte desde el primero mes. |

Unidades Temáticas

| Número | Nombre de la Unidad | Duración en Semanas |
|-----------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| 1 | GENERALIZE CPM 2016 to arbitrary Output Alphabet | 4 |
| Contenidos | Resultados de Aprendizajes de la Unidad | Referencias a la Bibliografía |
| Generalizar el algoritmo descrito en [CPM2016] de un alfabeto de output de tamaño $B=2$ a un alfabeto de output de tamaño general $B>2$ | Optimal Prefix Free Codes, Huffman algorithm, van Leeuwen's algorithm. | "Optimal Prefix Free Codes with Partial Sorting", $J\{\backslash\{e\}\}r\{\backslash\{e\}\}my$ Barbay, CPM 2016, 29:1--29:13. |

| Número | Nombre de la Unidad | Duración en Semanas |
|-------------------------------------------------------------------------------------------|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| 2 | IMPLEMENT CPM 2016 in Python | 4 |
| Contenidos | Resultados de Aprendizajes de la Unidad | Referencias a la Bibliografía |
| Implementar el algoritmo descrito en [CPM2016] en python y ejecutarlo sobre datos reales. | Python, Experimentación en Teoría de la Computación | "Optimal Prefix Free Codes with Partial Sorting", $J\{\backslash\{e\}\}r\{\backslash\{e\}\}my$ Barbay, CPM 2016, 29:1--29:13. |

| Número | Nombre de la Unidad | Duración en Semanas |
|------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|----------------------------------------------------------|
| 3 | GENERALIZE CPM 2016 to Alphabetic Search Trees | 4 |
| Contenidos | Resultados de Aprendizajes de la Unidad | Referencias a la Bibliografía |
| Generalizar (si se puede) las técnicas de análisis descritas en CPM 2016 en el problema similar de "Alphabetic Search Trees" | Alphabetic Search Trees | [1968-BOOK-TheArtOfComputerProgramming-Knuth], volume 3. |

| Número | Nombre de la Unidad | Duración en Semanas |
|--------------------------------------------------------------------------------------------------------------------|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4 | GENERALIZE CPM 2016 to Minimax trees | 4 |
| Contenidos | Resultados de Aprendizajes de la Unidad | Referencias a la Bibliografía |
| Generalizar (si se puede) las técnicas de análisis descritas en CPM 2016 en el problema similar de "Minimax trees" | Minimax Trees | [2009-CA-MinimaxTree sInLinearTimeWithApplications-GawrychowskiGagie] "Minimax Trees in Linear Time with Applications", Gawrychowski, Paweł and Gagie, Travis, 2009. |

| | |
|-----------------|---------------|
| Vigencia desde: | Autono 2017 |
| Elaborado por: | Jérémy Barbay |