

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO****Licenciatura en Ciencias de la Computación****Facultad de Ciencias**

Programa de la asignatura

**Denominación de la asignatura:*****Algoritmos de Apareamiento de Cadenas***

Clave:	Semestre: 6-8	Eje temático: Algoritmos			No. Créditos: 10
Carácter: Optativa		Horas		Horas por semana	Total de Horas
Tipo: Teórico-Práctica		Teoría:	Práctica:	7	112
		3	4		
Modalidad: Curso		Duración del programa: Semestral			

Asignatura con seriación indicativa antecedente: Autómatas y Lenguajes Formales; Análisis de Algoritmos**Asignatura con seriación indicativa subsecuente:** Ninguna**Objetivo general:**

Conocer y revisar algoritmos enfocados al proceso de cadenas de caracteres y árboles, que son importantes en diversas disciplinas como la minería de datos, bioinformática, proceso de lenguajes formales y naturales, corrección de ortografía, y muchas otras aplicaciones importantes.

Índice temático

Unidad	Temas	Horas	
		Teóricas	Prácticas
I	Herramientas para el apareamiento de cadenas	6	8
II	Autómatas para apareamiento	6	8
III	Búsqueda de cadenas con ventana deslizante	6	8
IV	Arreglos de sufijos	6	8
V	Estructuras para índices	6	8
VI	Índices	3	4
VII	Alineamientos	4.5	6
VIII	Patrones aproximados	6	8
IX	Periodos locales	4.5	6
Total de horas:		48	64
Suma total de horas:		112	

Contenido temático	
Unidad	Tema
I Herramientas para el apareamiento de cadenas	
I.1	Cadenas y autómatas.
I.2	Conceptos de combinatoria.
I.3	Algoritmos y complejidad.
I.4	Implementación de autómatas.
I.5	Técnicas básicas de apareamiento de cadenas.
I.6	Fronteras y tablas de prefijos.
II Autómatas para apareamiento	
II.1	El <code>\textit{trie}</code> de un diccionario.
II.2	Búsqueda de varias cadenas simultáneamente.
II.3	Implementación con función de fracaso.
II.4	Implementación con sucesor por omisión.
II.5	Localización de una cadena.
II.6	Localización de una cadena y función de fracaso.
II.7	Localización de una cadena y de sucesor por omisión.
III Búsqueda de cadenas con ventana deslizante	
III.1	Búsquedas sin memoria.
III.2	Tiempo de búsqueda.
III.3	Construcción de la tabla de sufijos buena.
III.4	Autómata para el mejor factor.
III.5	Búsqueda con una memoria.
III.6	Búsqueda con varias memorias.
III.7	Búsqueda en diccionario.
IV Arreglos de sufijos	
IV.1	Búsqueda de una lista de cadenas.
IV.2	Búsqueda con el prefijo común mayor.
IV.3	Preproceso de la lista.
IV.4	Ordenamiento de sufijos.
IV.5	Ordenamiento de sufijos con alfabetos de enteros acotados.
IV.6	Prefijos comunes de los sufijos.
V Estructuras para índices	
V.1	<i>Tries</i> de sufijos.
V.2	Árboles de sufijos.
V.3	Contextos para factores.
V.4	Autómata de sufijos.
V.5	Autómata de sufijos compacto.
VI Índices	
VI.1	Implementación de índices.
VI.2	Operaciones básicas.
VI.3	Transductor de posiciones.
VI.4	Repeticiones.

VI.5	Cadenas prohibidas.
VI.6	Motor de búsqueda.
VI.7	Búsqueda de rotaciones.
VII Alineamientos	
VII.1	Comparación de cadenas.
VII.2	Alineamiento óptimo.
VII.3	Sub-secuencia común mayor.
VII.4	Alineamientos con huecos.
VII.5	Alineamiento local.
VII.6	Heurísticas para alineamiento local.
VIII Patrones aproximados	
VIII.1	Apareamiento aproximado con comodines.
VIII.2	Apareamiento aproximado con diferencias.
VIII.3	Apareamiento aproximado con discordancias.
VIII.4	Apareamiento aproximado para patrones pequeños.
VIII.5	Heurística para apareamiento aproximado con diferencias.
IX Periodos locales	
IX.1	Factores de particionamiento.
IX.2	Detección de potencias.
IX.3	Detección de cuadrados.
IX.4	Ordenamiento de sufijos.

Bibliografía básica:

1. Crochemore, Hancart y Lecro, *Algorithms on Strings*, Cambridge University Press, traducción al inglés, 2007.
2. Gusfield, *Algorithms on Strings, Trees, and Sequences*, Cambridge University Press, décima impresión 2007.
3. Smyth, *Computing Patterns in Strings*, Addison-Wesley Longman, Reading Ma, 2003.

Bibliografía complementaria:

4. Apostolico y Galil, editores, *Pattern Matching Algorithms*, Oxford University Press, 1997.
5. Atallah, Editor, *Algorithms and Theory of Computation Handbook*, CRC Press, 1999.
6. Charras y Lecroq, *Handbook of Exact Matching Algorithms*, King's College London Publications, 2004.
7. Crochemore y Rytter, *Jewels of Stringology*, World Scientific Press, Singapore, 2002.
8. Navarro y Raffinot, *Flexible Pattern Matching in Strings -- Practical Online Search Algorithms for Texts and Biological Sequences*, Cambridge University Press, Cambridge, UK, 2002.
9. Smyth, *Computing Patterns in Strings*, Addison-Wesley Longman, Reading Ma, 2003.
10. Stephen, *String Searching Algorithms*, World Scientific Press, Singapore, 1994.
11. Además de múltiples artículos sobre el tema.

Sugerencias didácticas:		Métodos de evaluación:	
Exposición oral	(X)	Exámenes parciales	(X)
Exposición audiovisual	(X)	Examen final escrito	()
Ejercicios dentro de clase	(X)	Trabajos y tareas fuera del aula	(X)
Ejercicios fuera del aula	()	Prácticas de laboratorio	()
Seminarios	(X)	Exposición de seminarios por los alumnos	()
Lecturas obligatorias	(X)	Participación en clase	(X)
Trabajo de investigación	()	Asistencia	()
Prácticas de taller o laboratorio	(X)	Proyectos de programación	(X)
Prácticas de campo	()	Proyecto final	()
		Seminario	()
Otras: _____		Otras: _____	
Perfil profesiográfico: Egresado preferentemente de la Licenciatura en Ciencias de la Computación o matemático con especialidad en computación con amplia experiencia de programación. Es conveniente que posea un posgrado en la disciplina. Con experiencia docente.			