

Universidad Católica San Pablo
Facultad de Ingeniería y Computación
Escuela Profesional de
Ciencia de la Computación
SILABO



CS210T. Análisis y Diseño de Algoritmos (Obligatorio)

2010-1

1. DATOS GENERALES

1.1 CARRERA PROFESIONAL	:	Ciencia de la Computación
1.2 ASIGNATURA	:	CS210T. Análisis y Diseño de Algoritmos
1.3 SEMESTRE ACADÉMICO	:	5 ^{to} Semestre.
1.4 PREREQUISITO(S)	:	CS103O. Algoritmos y Estructuras de Datos. (4 ^{to} Sem) , CB203. Estadística y Probabilidades. (4 ^{to} Sem)
1.5 CARÁCTER	:	Obligatorio
1.6 HORAS	:	2 HT; 2 HP; 2 HL;
1.7 CRÉDITOS	:	4

2. DOCENTE

3. FUNDAMENTACIÓN DEL CURSO

Los algoritmos son pieza clave para la ciencia de la computación. El rendimiento de sistema de software depende sólo de dos cosas: a) La búsqueda de algoritmos y b) La eficiencia conveniente de varias capas de implementación. El diseño de buenos algoritmos es por otra parte crucial para el buen funcionamiento de todo sistema de software. Más aun, el estudio de algoritmos provee el buen entendimiento de la naturaleza del problema, así como también, técnicas independientes para la posible solución, independientemente de un lenguaje de programación, paradigma de programación, hardware de computador o cualquier otro aspecto de implementación.(Computing Curricula IEEE-CS & ACM).

4. SUMILLA

1. AL/Análisis Básico de Algoritmos.2. AL/Algoritmos Fundamentales.3. AL/Estrategias Algorítmicas.4. AL/Algoritmos Distribuidos.5. AL/Clases de Complejidad P y NP.

5. OBJETIVO GENERAL

- Permitir que el alumno pueda realizar el análisis y diseño de algoritmos eficientes para la solución de problemas complejos.
- Proveer al alumno de una serie de técnicas, de análisis y diseño para la evaluación e implementación de algoritmos.

6. CONTRIBUCIÓN A LA FORMACIÓN PROFESIONAL Y FORMACIÓN GENERAL

Esta disciplina contribuye al logro de los siguientes resultados de la carrera:

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. [Nivel Bloom: 6]
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. [Nivel Bloom: 5]
- h) Incorporarse a un proceso de aprendizaje profesional continuo. [Nivel Bloom: 3]
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. [Nivel Bloom: 3]
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. [Nivel Bloom: 5]

7. CONTENIDOS

UNIDAD 1: AL/Análisis Básico de Algoritmos.(12 horas)

Nivel Bloom: 4

OBJETIVO GENERAL

- Explicar el uso de las notaciones Big O , *Omega* Ω y *Theta* Θ para describir la cantidad de trabajo hecha por un algoritmo.
- Uso de notaciones Big O , *Omega* Ω y *Theta* Θ para determinar los límites asintóticos superior, inferior y el más próximo en tiempo y espacio en complejidad de algoritmos .
- Determinar la complejidad de tiempo y espacio de algoritmos simples.
- Deducir la relación de recurrencia que describe la complejidad de tiempo de algoritmos definidos recursivamente.
- Solucionar relaciones de recurrencia elemental.

CONTENIDO

- Análisis asintótico de límites en los casos promedio y superior.
- Identificar la diferencias entre el comportamiento entre el mejor, mediano y peor caso.
- Notación Big O , *little o*, *Omega* Ω y *Theta* Θ .
- Clases de complejidad estándar.
- Medición empírica de desempeño.
- Puntos de equilibrio entre tiempo vs espacio en algoritmos.
- Uso relaciones de recurrencia para el análisis de algoritmos recursivos.

Lecturas: [Kleinberg and Tardos, 2005], [Dasgupta et al., 2006], [Cormen et al., 2009], [Graham et al., 1994]

UNIDAD 2: AL/Algoritmos Fundamentales.(16 horas)	
Nivel Bloom: 5	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Implementar los algoritmos cuadráticos más comunes y los algoritmos de ordenamiento $O(N\log N)$. ▪ Diseñar e implementar una función de (<i>hash</i>) apropiada para una aplicación. ▪ Diseñar e implementar un algoritmo de resolución de colisiones para tablas de <i>hash</i>. ▪ Discutir la eficiencia computacional de los principales algoritmos de ordenamiento, búsqueda y (<i>hashing</i>). ▪ Discutir otros factores, además de la eficiencia computacional, que influyen en la elección de los algoritmos, tales como tiempo de programación, mantenimiento y el uso de patrones específicos de aplicación en los datos de entrada. ▪ Resolver problemas usando los algoritmos de grafos fundamentales, incluyendo búsqueda por amplitud y profundidad; caminos más cortos con uno y múltiples orígenes, cerradura transitiva, ordenamiento topológico y al menos un algoritmo de árbol de expansión mínima. ▪ Demostrar las siguientes capacidades: evaluar algoritmos, seleccionar una opción de un rango posible, proveer una justificación para tal elección e implementar el algoritmo.. 	<ul style="list-style-type: none"> ▪ Algoritmos numéricos simples. ▪ Búsqueda secuencial y binaria. ▪ Algoritmos cuadráticos de ordenamiento (selección, inserción). ▪ Algoritmos de tipo $O(N\log N)$ (Quicksort, heapsort, mergesort). ▪ Tablas de (<i>hash</i>) incluyendo estrategias de solución para las colisiones. ▪ Árboles de búsqueda binaria. ▪ Representación de grafos (Listas y Matrices de adyacencia). ▪ Recorridos por amplitud y profundidad. ▪ El algoritmo del camino más corto (algoritmos de Dijkstra y Floyd). ▪ Cerradura transitiva (algoritmo de Floyd). ▪ Árbol de expansión mínima (algoritmos de Kruskal y Prim). ▪ Ordenamiento Topológico.
Lecturas: [Kleinberg and Tardos, 2005], [Dasgupta et al., 2006], [Cormen et al., 2009]	

UNIDAD 3: AL/Estrategias Algorítmicas.(24 horas)	
Nivel Bloom: 5	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Describir las desventajas de los algoritmos de fuerza bruta. ▪ Para cada una de las diferentes clases de algoritmos (fuerza bruta, voraces, dividir y conquistar, <i>Backtracking</i>, <i>Branch-and-bound</i> y heurísticos), identificar un ejemplo del comportamiento humano cotidiano que ejemplifique el concepto básico. ▪ Implementar un algoritmo voraz para resolver apropiadamente un problema. ▪ Implementar un algoritmo de divide y vencerás para solucionar apropiadamente un problema. ▪ Utilizar <i>Backtracking</i> para solucionar problemas tal como el de navegación en un laberinto. ▪ Describir varios métodos de solución de problemas heurísticos. ▪ Utilizar emparejamiento de patrones para analizar subcadenas. ▪ Utilizar aproximación numérica para resolver problemas matemáticos, tal como el de encontrar las raíces de un polinomio. 	<ul style="list-style-type: none"> ▪ Algoritmos de fuerza bruta (<i>brute-force</i>). ▪ Algoritmos voraces (<i>greedy</i>). ▪ Divide y vencerás. ▪ <i>Backtracking</i>. ▪ <i>Branch-and-bound</i>. ▪ Heurísticos. ▪ Emparejamiento de patrones y algoritmos de cadenas/texto. ▪ Algoritmos de aproximación numérica.
Lecturas: [Kleinberg and Tardos, 2005], [Dasgupta et al., 2006], [Cormen et al., 2009]	

UNIDAD 4: AL/Algoritmos Distribuidos.(4 horas)	
Nivel Bloom: 4	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Explicar el paradigma distribuido. ▪ Explicar un algoritmo distribuido simple. ▪ Determinar cuando usar los algoritmos de consenso o elección. ▪ Distinguir entre relojes físicos y lógicos. ▪ Describir el ordenamiento relativo de eventos en un algoritmo distribuido. 	<ul style="list-style-type: none"> ▪ Consenso y elección. ▪ Detección de finalización. ▪ Tolerancia a fallas. ▪ Estabilización.
Lecturas: [Kleinberg and Tardos, 2005], [Dasgupta et al., 2006], [Cormen et al., 2009]	

UNIDAD 5: AL/Clases de Complejidad P y NP.(4 horas)	
Nivel Bloom: 5	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Definir las clases P y NP. ▪ Explicar el significado de la NP-Complejidad. ▪ Probar que un problema es NP-completo reduciendo un problema NP-Completo clásico conocido a éste. 	<ul style="list-style-type: none"> ▪ Definición de las clases P y NP. ▪ NP-complejidad (El teorema de Cook). ▪ Problemas NP-completos estándares. ▪ Técnicas de reducción.
Lecturas: [Kleinberg and Tardos, 2005], [Dasgupta et al., 2006], [Cormen et al., 2009]	

8. METODOLOGÍA
<p>El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.</p> <p>El profesor del curso presentará demostraciones para fundamentar clases teóricas.</p> <p>El profesor y los alumnos realizarán prácticas</p> <p>Los alumnos deberán asistir a clase habiendo leído lo que el profesor va a presentar. De esta manera se facilitará la comprensión y los estudiantes estarán en mejores condiciones de hacer consultas en clase.</p>

9. EVALUACIONES
<p>Evaluación Permanente 1 : 20 %</p> <p>Examen Parcial : 30 %</p> <p>Evaluación Permanente 2 : 20 %</p> <p>Examen Final : 30 %</p>

Referencias

- [Cormen et al., 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition.
- [Dasgupta et al., 2006] Dasgupta, S., Papadimitriou, C., and Vazirani, U. (2006). *Algorithms*. McGraw-Hill Education.
- [Graham et al., 1994] Graham, R. L., Knuth, D. E., and Patashnik, O. (1994). *Concrete Mathematics*. Addison Wesley Iberoamericana.
- [Kleinberg and Tardos, 2005] Kleinberg, J. and Tardos, E. (2005). *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc.