CIENCIA DE LA COMPUTACIÓN

ANÁLISIS Y DISEÑO DE ALGORITMOS

4 CRÉDITOS

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

CE2A

ÍNDICE

UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA
SILABO 2021-1

1. ASIGNATURA

CS2102 - Análisis y Diseño de Algoritmos

2. DATOS GENERALES

2.1 Ciclo: 5
2.2. Créditos: cuatro (4) créditos
2.3 Horas de teoría: dos (2) semanales
2.4 Horas de práctica: cuatro (4) semanales
2.5 Duración del período: dieciséis (16) semanas
2.6. Condición:
- Obligatorio para Ciencia de la Computación Modalidad: Virtual

2.7. Modalidad: Virtual
2.8. Requisitos:
- CS2100- Algoritmos y Estructura de Datos

3. PROFESORES

3.1. Profesor coordinador del curso

Juan Gutiérrez (jgutierreza@utec.edu.pe)
Horario de atención: Martes 5pm-6pm

3.2. Profesor(es) instructor(es) del curso

Ronald Ubel Adolfo Gonzales Vega, (rgonzalesveg@utec.edu.pe)
Horario de atención: previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Un algoritmo es un conjunto finito de reglas o instrucciones que permiten resolver un problema. El estudio teórico de algunos aspectos como su rendimiento, su tiempo de ejecución o su espacio utilizado nos permite analizar y definir qué tan apropiado es un determinado algoritmo para resolver un problema en específico. Estos algoritmos detrás de las soluciones a diversos problemas han promovido el desarrollo de las tecnologías que se usan día a día en diversos campos como economía, geología, exploración espacial, medicina, biología, entre otros.

Podemos definir la Ciencia de la Computación como el estudio de algoritmos. En este curso se presentan las técnicas usadas en el análisis y diseño de algoritmos con el propósito de aprender y aplicar los principios fundamentales

para el diseño e implementación de métodos computacionales en la resolución de problemas.

Este curso es de vital importancia para el desarrollo profesional del estudiante, ya que proporciona conceptos clave para la Ciencia de la Computación, los cuales son tratados de manera mucho más formal que en otros cursos. Con estos conceptos el estudiante será capaz de desarrollar software mucho más sólido en la industria, ya que conocerá a fondo los principios sobre los cuales descansa la Ciencia de la Computación.

5. OBJETIVOS

Sesión 1. Desarrollar la capacidad para evaluar la complejidad y calidad de algoritmos propuestos para un determinado problema.

Sesión 2. Desarrollar la habilidad para resolver problemas de divisón y conquista usando los principios de diseño de algoritmos aprendidos.

Sesión 3. Conocer el análisis de la estructura heap en cuanto a tiempo y espacio. Aplicar la estructura heap en situaciones prácticas como ordenamientos y uso de fila de prioridades.

Sesión 4. Conocer el concepto de tiempo esperado de ejecución. Analizar el tiempo esperado usando conceptos de probabilidades. Aplicar el análisis al quicksort.

Sesión 5. Conocer la diferencia entre un diseño de programación dinámica para un problema y un diseño usando recursividad. Aplicar técnicas de programación dinámica en problemas específicos

Sesión 6. Conocer los conceptos básicos de la estrategia voraz. Aprender a demostrar la correctitud de un algoritmo voraz. Diseñar algoritmos voraces para problemas específicos.

Sesión 7. Conocer los beneficios de hacer un análisis amortizado. Aprender las diferentes técnicas para hacer dicho análisis así como su aplicación en distintos problemas

Sesión 8. Analizar los distintos algoritmos en grafos con las técnicas estudiadas a lo largo del curso. Aplicar algoritmos en grafos para distintos problemas.

6. COMPETENCIAS Y CRITERIOS DE DESEMPEÑO

Los criterios de desempeño que se van a trabajar en este curso son:

1.1. Aplica conocimientos de matemáticas apropiados para la solución de problemas definidos y sus requerimientos en la disciplina del programa.(nivel 3)

1.3. Aplica conocimientos de computación apropiados para la solución de problemas definidos y sus requerimientos en la disciplina del programa (nivel 3)

2.4: Resuelve problemas de computación y otras disciplinas relevantes en el dominio (nivel 2).

5.1. Trabaja eficazmente en equipo (nivel 2)

8.2. Se compromete con la práctica profesional de la computación (nivel 1)

7. RESULTADOS DE APRENDIZAJE

Al final del curso Análisis y diseño de algoritmos se espera que el estudiante sea capaz de:

RA1. Evaluar el tiempo de ejecución de un algoritmo a partir del análisis de recurrencias y teorema maestro.

RA2. Diseñar nuevos algoritmos, con restricciones de tiempo y memoria, utilizando las técnicas vistas en clase

RA3. Usar las técnicas de división y conquista, algoritmos voraces y programación dinámica para resolver ejercicios de carácter algorítmico

RA4. Implementar software de manera grupal y colaborativa, donde cada integrante tenga a su cargo un módulo

RA5. Reconocer la importancia del análisis de algoritmos en la industria, así como su relación con el mundo real

8. TEMAS

1. **Introducción**
   1.1. Algoritmos, problema de ordenación, comparación intuitiva del tiempo de ejecución entre dos algoritmos
   1.2. Sumatorias, fórmulas y propiedades básicas: linealidad, series aritméticas, suma de cuadrados y cubos, serie geométrica, serie armónica, series telescópicas
   1.3. Acotando sumatorias: inducción, acotando términos, dividiendo sumatorias
   1.4. Conceptos básicos de grafos y árboles
   1.5. Insertion sort: correctitud y análisis
   1.6. Crecimiento de funciones: notación Theta, notación O-grande, notación Gamma, notación Omega, notación o-pequeña

2. **División y conquista**
   2.1. Ingredientes básicos del método de división y conquista
   2.2. Análisis del Mergesort: correctitud y análisis del tiempo de ejecución
   2.3. Recurrencias: resolución por expansión, verificación por inducción, teorema maestro
   2.4. Problema del subarreglo máximo
   2.5. Multiplicación de números naturales (algoritmo de Karatsuba)
   2.6. Multiplicación de matrices (algoritmo de Strassen)
   2.7. Conteo de inversiones

3. **Heap**
   3.1. Definición de heap, propiedades básicas de heap
   3.2. Construcción de max-heap: max-heapify y build-max-heap
   3.3. El algoritmo heapsort
   3.4. Filas de prioridades

4. **Análisis Probabilístico y Quicksort**
   4.1. Repaso de probabilidades:variable aleatoria, función de probabilidad, valor esperado
   4.2. El problema de contratación (Hiring problem)
   4.3. El algoritmo Quicksort. Análisis de tiempo del Quicksort: peor caso, mejor caso y caso promedio

5. **Programación Dinámica**
   5.1. Conceptos básicos de Programación dinámica y memorización
   5.2. Números de Fibonacci
   5.3. Coeficientes binomiales
   5.4. Intervalos disjuntos
   5.5. Subsecuencia creciente máxima
   5.6. Subset sum y mochila
   5.7. Partición lineal justa

6. **Algoritmos voraces (Greedy)**
   6.1. Conceptos básicos de la estrategia voraz
   6.2. Técnicas para demostrar la correctitud de un algoritmo voraz
   6.3. Intervalos disjuntos
   6.4. Planificación de tareas
   6.5. Problema de la mochila fraccionaria
   6.6. Código de Huffman

7. **Análisis amortizado**

7.1. Problema de operaciones en pilas. Problema del contador binario
7.2. Análisis agregado. Método de recargas
7.3. Heurística MTF (move to front). Splay trees

8. Algoritmos en grafos

8.1. Representaciones: matrices de adyacencia y listas de adyacencia
8.2. Búsqueda en profundidad
8.3. Búsqueda en largura
8.4. Caminos mínimos en grafos: algoritmo de Dijkstra, algoritmo de Bellman-Ford
8.5. Caminos mínimos entre todos los pares: algoritmos basados en multiplicación de matrices, algoritmo de Floyd-Warshall, algoritmo de Johnson

9. PLAN DE TRABAJO

9.1. Metodología

Se fomenta la participación individual y en equipo para exponer las ideas de los estudiantes, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso. A lo largo del curso se proporcionan diferentes lecturas, las cuales serán evaluadas. El uso de herramientas online permitirá al alumno acceder a la información del curso, interactuar fuera del aula con el profesor y con los otros estudiantes.

9.2. Sesiones de teoría

Las sesiones de teoría se llevan a cabo en clases magistrales online donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

9.3. Sesiones de práctica (laboratorio o taller)

Las sesiones prácticas/laboratorio se desarrollarán a través de una metodología activa generando el aprendizaje práctico por parte del estudiante.
Las sesiones de práctica se caracterizan por el desarrollo de problemas tanto de demostración de propiedades como de diseño de nuevos algoritmos.

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos para que permitan evaluar el desempeño de los alumnos.

10. SISTEMA DE EVALUACIÓN

EVALUACIÓN	**TEORÍA**	**PRÁCTICA Y/O LABORATORIO**
	Examen **E1** (20%) Examen **E2** (20%) Examen **E3** (20%) Examen **E4** (20%)	Proyecto **P1** (20%)
	80%	20%
	100%	

Las rúbricas que permitirán medir las actividades más significativas del curso y que, además se relacionan con la evaluación de las competencias del estudiante son: enlace

11. SESIONES DE APOYO O TUTORÍAS

Este apartado permite formalizar los espacios de apoyo a los estudiantes y que éstos tengan la atención NECESARIA y el tiempo disponible para presentar sus dudas y consultas acerca del curso:

Semana	Fecha/ Hora	Tema a tratar	Objetivos de la sesión
2	27/04/2021 5:00 PM – 6:00 PM	Evaluaciones	Resolver dudas sobre sistema de evaluación y temas a tratar en el curso
4	11/05/2021 5:00 PM –6:00 PM	Tópicos del curso	Resolver dudas sobre primeras 3 semanas
6	25/05/2021 5:00 PM – 6:00 PM	Tópicos del curso	Resolver dudas sobre primeras 5 semanas
8	08/06/2021 5:00 PM – 6:00 PM	Parcial	Reclamos examen parcial
10	22/06/2021 5:00 PM – 6:00 PM	Tópicos del curso	Resolver dudas sobre temas tratados en clase

12	06/07/2021 5:00 PM – 6:00 PM	Proyecto	Resolver dudas sobre entrega del proyecto
14	20/07/2021 5:00 PM – 6:00 PM	Final	Consultas sobre ultimas notas y examen final

12. REFERENCIAS BIBLIOGRÁFICAS

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2009). *Introduction to algorithms*. MIT press.
- S. Dasgupta, C.H. Papadimitriou, U.V. Vazirani, Algorithms, McGraw-Hill, 2006
- Jon Kleinberg, Éva Tardos, Algorithm Design, Addison-Wesley, 2005.
- Graham, R. L., Knuth, D. E., & Patashnik, O. (1994). *Concrete mathematics: A foundation for computer science*. Reading, Mass: Addison-Wesley.
- Sedgewick, R., & Flajolet, P. (2013). *An introduction to the analysis of algorithms*. Pearson Education India.
- Sedgewick, R., & Wayne, K. (2011). *Algorithms*. Addison-wesley professional.
- Knuth, D. E., & Knuth, D. F. (1973). *The art of computer programming: Fundamental algorithms* (Vol. 3). Addison Wesley Publishing Company. Jon Bentley, Programming Pearls, Addison-Wesley, 1986