CIENCIA DE LA COMPUTACIÓN

ANÁLISIS Y DISEÑO DE ALGORITMOS

4 CRÉDITOS

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

CE2A

REGLAS INTEGRIDAD ACADÉMICA

Todo estudiante matriculado en una asignatura de la Universidad de Ingeniería y Tecnología tiene la obligación de conocer y cumplir las reglas de integridad académica, cuya lista a continuación es de carácter enunciativo y no limitativo, ya que el/la docente podrá dar mayores indicaciones:

1. La copia y el plagio son dos infracciones de magnitud muy grave en la Universidad de Ingeniería y Tecnología (UTEC) conforme a lo establecido en el Reglamento de Disciplina de los Estudiantes. Tienen una sanción desde 2 semestres de suspensión hasta la expulsión.
2. Si se identifica la copia o plagio en evaluaciones individuales, el/la docente puede proceder a anular la evaluación.
3. Si la evaluación es personal o grupal-individual, la interacción entre equipos o compañeros se considera copia o plagio, según corresponda. Si la evaluación calificada no indica que es grupal, se presume que es individual.
4. La copia, plagio, el engaño y cualquier forma de colaboración no autorizada no serán tolerados y serán tratados de acuerdo con las políticas y reglamentos de la UTEC, implicando consecuencias académicas y sanciones disciplinarias.
5. Aunque se alienta a los estudiantes a discutir las tareas y trabajar juntos para desarrollar una comprensión más profunda de los temas presentados en este curso, no se permite la presentación del trabajo o las ideas de otros como propios. No se permite el plagio de archivos informáticos, códigos, documentos o dibujos.
6. Si el trabajo de dos o más estudiantes es sospechosamente similar, se puede aplicar una sanción académica a todos los estudiantes, sin importar si es el estudiante que proveyó la información o es quien recibió la ayuda indebida. En ese sentido, se recomienda no proveer el desarrollo de sus evaluaciones a otros compañeros ni por motivos de orientación, dado que ello será considerado participación en copia.
7. El uso de teléfonos celulares, aplicaciones que permitan la comunicación o cualquier otro tipo de medios de interacción entre estudiantes está prohibido durante las evaluaciones o exámenes, salvo que el/la docente indique lo contrario de manera expresa. Es irrelevante la razón del uso del dispositivo.
8. En caso exista algún problema de internet durante la evaluación, comunicarse con el/la docente utilizando el protocolo establecido. No comunicarse con los compañeros dado que eso generará una presunción de copia.
9. Se prohíbe tomar prestadas calculadoras o cualquier tipo de material de otro estudiante durante una evaluación, salvo que el/la docente indique lo contrario.
10. Si el/la docente encuentra indicios de obtención indebida de información, lo que también implica no cumplir con las reglas de la evaluación, tiene la potestad de anular la prueba, advertir al estudiante y citarlo con su Director de Carrera. Si el estudiante no asiste a la citación, podrá ser reportado para proceder con el respectivo procedimiento disciplinario. Una segunda advertencia será reportada para el inicio del procedimiento disciplinario correspondiente.
11. Se recomienda al estudiante estar atento/a a los datos de su evaluación. La consignación de datos que no correspondan a su evaluación será considerado indicio concluyente de copia.

--
Fecha de actualización: 21/07/2021
Revisado y aprobado por el Centro de Excelencia en Enseñanza y Aprendizaje y la Dirección de Ciencia de la Computación

ÍNDICE

--
Fecha de actualización: 21/07/2021
Revisado y aprobado por el Centro de Excelencia en Enseñanza y Aprendizaje y la Dirección de Ciencia de la Computación

UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA
SILABO 2022-2

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 Ingeniería Ambiental

2.7 **Modalidad:** Presencial
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: previa coordinación con el profesor

3.2 Profesor(es) instructor(es) del curso
Juan Gutiérrez (jgutierreza@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 división 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

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. Resuelve problemas y hace demostraciones de manera grupal y colaborativa.

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. Insertion sort: correctitud y análisis
1.5. 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 memoización
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

6.1. Intervalos disjuntos
6.2. Conceptos básicos de la estrategia voraz
6.3. Técnica para demostrar la correctitud de un algoritmo voraz
6.4. Problema de la mochila fraccionaria
6.5. Árboles de Huffman

7. Análisis amortizado

7.1. Problema de operaciones en pilas. Problema del contador binario
7.2. Análisis agregado.
7.3. Método de recargas.
7.4. Método del potencial

8. Ordenación en tiempo lineal
8.1. Cotas inferiores para ordenar
8.2. Counting sort
8.3. Radix sort
8.4. Bucket sort

9. Algoritmos en grafos
9.1. Caminos mínimos en grafos: algoritmo de Dijkstra, algoritmo de Bellman-Ford
9.2. Caminos mínimos entre todos los pares: algoritmos basados en multiplicación de matrices, algoritmo de Floyd-Warshall, algoritmo de Johnson
9.3. Flujos en grafos

9. PLAN DE TRABAJO

9.1 Metodología

En el curso se aplicarán las metodologías de clase invertida, aprendizaje cooperativo, aprendizaje basado en problemas, activa y expositiva. Estas metodologías aumentan el interés del estudiante y promueven su compromiso en el aprendizaje.

9.2 Sesiones de teoría

Las sesiones de teoría se llevan a cabo en clases magistrales 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

	TEORÍA	PRÁCTICA Y/O LABORATORIO
EVALUACIÓN *Para aprobar el curso es necesario aprobar tanto la parte teórica como la parte práctica	3 exámenes **(E)**: Examen **E1** (25%) Examen **E2** (25%) Examen **E3** (25%)	Evaluación continua **C** (25%)
	75%	25%
	100%	

Examen – rúbrica

11. 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