

Fundamentos de Algoritmos

Profs. Agustín Gravano y Hernán Czemerinski

Resumen y Objetivos del Curso

Este curso introduce conceptos elementales de programación. Al finalizar el curso, se espera que el alumno pueda especificar y resolver problemas de tamaño pequeño a mediano mediante la programación de algoritmos. En particular, se apunta a que los alumnos dominen elementos básicos de programación, tipos de datos básicos y avanzados, procesamiento de archivos de texto, especificación de problemas, testing de programas, y estimación de la complejidad temporal de programas.

Métodos y Materiales

Se emplearán una variedad de métodos de enseñanza y materiales. Las clases consistirán de difusión teórica y resolución de problemas. Habrá guías de problemas para ejercitar los contenidos de cada clase, que serán de realización obligatoria. La bibliografía del curso será en inglés, por lo que se requiere al menos de un nivel intermedio de lectura y comprensión en ese idioma. Emplearemos también términos técnicos propios de la jerga de programación que típicamente se expresan en inglés. Si bien se utilizará extensivamente el lenguaje de programación Python, no se requieren conocimientos previos en el manejo del mismo.

Evaluación

La nota estará basada en el siguiente esquema de ponderación:

- | | |
|--|-----|
| • Participación en clase, asistencia y puntualidad | 10% |
| • Trabajos prácticos grupales | 50% |
| • Examen final | 40% |

Participación en clase

La participación en clase es una parte importante del curso, por lo que la misma será evaluada en forma continua. La evaluación de la participación estará basada en cuán al día están los alumnos con los temas de la materia, en particular respecto de la ejercitación obligatoria. Dado que los contenidos de esta materia se acumulan semana a semana, es fundamental

mantenerse al día para poder participar interactivamente del dictado de clases, y así optimizar la transmisión de conocimientos.

Trabajos prácticos (TPs)

Habrán dos trabajos prácticos grupales. A los efectos del curso, deberán formarse grupos de tres personas. Para los TPs no se admite colaboración entre miembros de distintos grupos. Es requisito para aprobar el curso la aprobación de los TPs. Los trabajos deberán subirse a la página de la materia en el campus virtual, en el formato que se indique.

Examen final

El examen es individual. Es requisito para aprobar el curso la aprobación del examen final, para lo que se requiere un total de 60/100 puntos.

Bibliografía

- “Aprenda a Pensar como un Programador (con Python)”, Green Tea Press, 2002, <https://argentinaenpython.com/quiero-aprender-python/aprenda-a-pensar-como-un-programador-con-python.pdf>
- “The Python Tutorial”, <https://docs.python.org/3/tutorial/>.
- T.H. Cormen y otros, "Introduction to Algorithms", tercera edición, MIT Press, 2009.

Contenidos del Curso

Modulo I: Elementos Básicos de Programación

Sesión #1: Introducción a la programación imperativa

- Definición de programa: secuencia finita de instrucciones.
- Concepto de estado de un programa y transformación de estados.
- Variable, memoria, expresiones, tipos de datos básicos: bool, int, float.
- Instrucción, asignación.

Sesión #2: Estructuras de control y funciones

- Condicional: expresiones booleanas, lógica trivaluada y evaluación lazy.

- Ciclo: ciclos anidados, alcance de las variables.
- Funciones. Argumentos y valor de retorno.

Módulo II: Tipos de Datos

Sesión #3: Procesamiento de listas y archivos

- Tipo lista: constructores y operaciones. Listas por comprensión. Iteración.
- Secuencias de caracteres. Tipo string.
- Lectura y escritura de archivos de texto.

Sesión #4: Tipos de datos avanzados

- Tipos conjunto, diccionario, árbol y grafo. Ejemplos de uso.
- Definición de tipos de datos específicos. Partes pública y privada.

Módulo III: Análisis de Programas

Sesión #5: Complejidad de algoritmos

- Complejidad algorítmica. Análisis del tiempo de ejecución de un programa en función del tamaño de la entrada.
- Algoritmos clásicos de búsqueda y ordenamiento. Búsqueda lineal y binaria. Bubble sort, insertion sort y selection sort.

Sesión #6: Verificación de algoritmos

- Conceptos básicos de verificación formal de programas respecto de una especificación.
- Testing de algoritmos. Testing de unidad. Elección de casos de test.