Aprobado por resolución N°113 del CFI de fecha 04.07.2017

FACULTAD DE INGENIERIA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Programa de Programación 3

1. NOMBRE DE LA UNIDAD CURRICULAR

Programación 3

2. CRÉDITOS

15 créditos

3. OBJETIVOS DE LA UNIDAD CURRICULAR

1. Que el estudiante conozca algoritmos clásicos que constituyen una base para la resolución de problemas en computación y que sea capaz de aplicarlos para la resolución de problemas concretos.
2. Que el estudiante domine técnicas generales de diseño de algoritmos y que sea capaz de aplicarlas y de realizar implementaciones en un lenguaje de programación.
3. Que el estudiante desarrolle habilidades de análisis que le permitan razonar con rigurosidad sobre cualidades como la corrección y los requerimientos de recursos de cómputo de diferentes algoritmos.
4. Que el estudiante sea capaz de analizar rigurosamente problemas algorítmicos para identificar limitaciones de cómputo inherentes a cada problema, reconociendo diferentes clases de complejidad.

4. METODOLOGÍA DE ENSEÑANZA

El contenido teórico del temario se discutirá en clases teóricas, a razón de 2 clases por semana de 2 hs. cada una, siguiendo el contenido de la bibliografía recomendada.

Se propondrán ejercicios de práctico, para resolución en domicilio, con el objetivo de afianzar los conocimientos teóricos y contribuir al cumplimiento de los objetivos de la unidad curricular. Asimismo se hará disponible a los estudiantes un cronograma sugerido de avance en la resolución de estos ejercicios.

Algunos de las técnicas de diseño de algoritmos estudiadas serán puestas en práctica a través de trabajos de laboratorio que involucrarán la implementación de algoritmos en máquina.

Cada estudiante participará de un encuentro de monitoreo semanal, de 1 hora de duración cada uno, donde se hará un seguimiento de su grado de avance y podrá consultar dudas sobre ejercicios prácticos y de laboratorio; algunas de las horas de monitoreo serán destinadas a instancias de evaluación.

Se estiman 10 horas de dedicación semanal adicional del estudiante para la realización de prácticos y laboratorios.

5. TEMARIO

1. **Análisis de algoritmos.**
   Análisis de corrección y de complejidad en peor caso y en media.
   Repaso de notación asintótica O, Ω, y Θ.
   Análisis de algoritmos recursivos.
   Análisis de amortización de tiempo de ejecución.
2. **Algoritmos fundamentales sobre grafos (puede tener variaciones en cada edición).**
   Estructuras de datos para la representaciones de grafos.
   Exploración de grafos.
   Búsqueda del camino más corto (algoritmo de Dijkstra).
   Identificación de componentes conexas y fuertemente conexas.
   Verificación de condición de bipartito.
   Construcción de árboles de cubrimiento mínimo (algoritmos de Prim y Kruskal).
   Construcción de un orden topológico en grafos dirigidos acíclicos.
   Aplicaciones.
3. **Algoritmos de ordenamiento.**
   Ejemplos de algoritmos clásicos como Quicksort, Heapsort y Mergesort.
   Cota inferior asintótica para la complejidad de algoritmos de ordenamiento.
4. **Otros algoritmos clásicos (puede tener variaciones en cada edición).**
   Algoritmos sobre cadenas de texto, como por ejemplo alineamiento de secuencias.
   Algoritmos sobre conjuntos: Union-Find.
5. **Técnicas de diseño de algoritmos.**
   Búsqueda exhaustiva.
   Divide y vencerás.
   Algoritmos ávidos (Greedy).
   Programación dinámica.
   Algoritmos de mejoramiento iterativo.

6. **Algoritmos probabilísticos.**

   La aleatoriedad interna como herramienta para evitar hipótesis probabilísticas sobre las entradas en análisis de rendimiento promedio (por ejemplo Quicksort aleatorio).

   Algoritmos que obtienen resultados correctos con alta probabilidad (pero no con certeza).

   Análisis de algoritmos aleatorios.

7. **Problemas intratables**

   Clases P y NP.

   La clase de problemas NP-completos. Ejemplos.

   Reducción de problemas.

   Técnicas para atacar problemas intratables.

6. BIBLIOGRAFÍA

Tema	Básica	Complementaria
Todos los temas	(1,2,3)	

6.1 Básica

1. Kleinberg, J., & Tardos, E. (2006). Algorithm design. Boston: Pearson/Addison-Wesley.
2. Cormen, T. H., & Cormen, T. H. (2001). Introduction to algorithms. Cambridge, Mass: MIT Press.
3. G. Brassard & P. Bratley. (1998). Fundamentos de Algoritmia. Madrid: Prentice Hall. ISBN 84-89660-00-X. (en biblioteca de Facultad de Ingeniería).

7. CONOCIMIENTOS PREVIOS EXIGIDOS Y RECOMENDADOS

7.1 Conocimientos Previos Exigidos: Conocimientos de programación estructurada. Estructuras de datos clásicas: colas, pilas, colas de prioridad, listas, tablas de dispersión y árboles. Conocimientos de matemática discreta (incluyendo inducción sobre estructuras y teoría de grafos).

7.2 Conocimientos Previos Recomendados: Nociones elementales de probabilidad (variables aleatorias discretas, probabilidad conjunta y condicional, esperanza).

ANEXO A
Para todas las Carreras

A1) INSTITUTO

Instituto a cargo de la unidad curricular: **Instituto de Computación**

A2) CRONOGRAMA TENTATIVO

Semana 1	Introducción. Repaso de análisis de algoritmos.
Semana 2	Ejemplo de análisis de un algoritmo sencillo (Apareamiento Estable).
Semana 3	Algoritmos fundamentales sobre grafos.
Semana 4	Algoritmos fundamentales sobre grafos.
Semana 5	Búsqueda exhaustiva. Algoritmos ávidos.
Semana 6	Árboles de cubrimiento. Estructura Union-Find.
Semana 7	Divide y vencerás.
Semana 8	Divide y vencerás. Algoritmos de ordenamiento.
	PERÍODO DE PARCIALES
Semana 9	Programación dinámica.
Semana 10	Programación dinámica. Alineación de secuencias.
Semana 11	Algoritmos de mejoramiento iterativo.
Semana 12	Problemas intratables.
Semana 13	Problemas intratables.
Semana 14	Algoritmos probabilísticos.
Semana 15	Repaso general.

A3) MODALIDAD DEL CURSO Y PROCEDIMIENTO DE EVALUACIÓN

La unidad curricular se evalúa por medio de trabajos de laboratorio de carácter eliminatorio (al menos uno), controles de monitoreo y pruebas parciales.

Se presenta a continuación el esquema de evaluación:

El nivel mínimo de suficiencia en los trabajos de laboratorio es eliminatorio.

Los controles de monitoreo son instancias de evaluación individual que tienen lugar durante algunas de las sesiones de monitoreo, en semanas preestablecidas en el cronograma del curso que se darán a conocer con antelación. Estas evaluaciones consistirán en la respuesta

de preguntas teóricas y/o resolución de ejercicios prácticos iguales o muy similares a los trabajados durante el desarrollo del curso. La cantidad de controles de monitoreo puede variar con cada edición, pero estará en el entorno de 4.

Los parciales son instancias de evaluación teórico-práctica que se realizan con el objetivo de evaluar los conocimientos adquiridos por el estudiante y su capacidad de incorporarlos en la solución de problemas. A diferencia de los controles de monitoreo, los parciales pueden incluir problemas con planteos sustancialmente diferentes a los ejercicios prácticos, aunque se resuelven usando las mismas herramientas y tienen un nivel de dificultad similar.

Tanto en los parciales como en los controles de monitoreo el estudiante acumula puntos que, junto con el resultado de los trabajos de laboratorio, determinan alguno de los siguientes resultados:

- Exoneración del examen final: el estudiante aprueba la unidad curricular.
- Suficiencia en el curso: el estudiante queda habilitado a rendir el examen.
- Insuficiencia en el curso: el estudiante reprueba el curso.

Las pruebas parciales representan en conjunto un total de 100 puntos, mientras que los controles de monitoreo representan en conjunto un total de 12 puntos. Se presentan a continuación las condiciones que deben alcanzarse para obtener la exoneración y para obtener la suficiencia en el curso; si no se cumplen estos últimos requisitos el resultado será el de insuficiencia en el curso.

- Requisitos para exoneración:

- llegar al nivel mínimo en los trabajos de laboratorio,
- reunir al menos 60 puntos en total,
- reunir al menos 6 puntos en controles de monitoreo.

- Requisitos para suficiencia en el curso:

- llegar al nivel mínimo en los trabajos de laboratorio,
- reunir al menos 25 puntos en total,
- reunir al menos 4 puntos en controles de monitoreo.

Dependiendo de las condiciones de dictado del curso, el trabajo de laboratorio se evalúa según las opciones aprobado/no aprobado, o con puntaje diferenciado. En el segundo caso, aquellos estudiantes que superen cierto umbral de puntos de laboratorio podrán acumular, junto con los controles de monitoreo, puntos adicionales para la exoneración o aprobación del curso; el total de puntos acumulados entre laboratorio y controles de monitoreo no excederá la cantidad de 12 puntos en total.

A4) CALIDAD DE LIBRE

Esta unidad curricular no adhiere a la resolución del consejo sobre condición de libre.

A5) CUPOS DE LA UNIDAD CURRICULAR

No tiene cupos.

ANEXO B para la carrera Ingeniería en Computación (plan 97)

B1) ÁREA DE FORMACIÓN

Programación

B2) UNIDADES CURRICULARES PREVIAS

Para el Curso: Cursos de Programación 2.
Curso de Probabilidad y Estadística
Exámenes de Programación 1 y Matemática Discreta 1.

Para el Examen: Curso de Programación 3.