

UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA		1 5
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
ALGORITMOS Y COMPLEJIDAD	CÓDIGO: 5523	
	ÁREA N°: II	

<b>CARRERAS</b> Ingeniería en Sistemas de Información					
<b>PROFESOR RESPONSABLE:</b> Dr. Pablo R. Fillotrani – Profesor Asociado con Dedicación Exclusiva					
<b>CARGA HORARIA</b>	<b>Teoría</b> 64	<b>Práctica</b> 50	<b>Laboratorio</b> 14	<b>CANTIDAD DE SEMANAS</b>	16
<b>CORRELATIVAS</b>					
PARA CURSAR LA MATERIA			PARA APROBAR LA MATERIA		
<b>APROBADAS</b> Tecnología de Programación	<b>CURSADAS</b> Lógica para Cs. de la Computación.	<b>APROBADAS</b> Lógica para Cs. de la Computación.	<b>CURSADAS</b>		
<b>DESCRIPCIÓN</b>					
<p>El objetivo específico de esta materia es afianzar el uso de técnicas de análisis de algoritmos que permitan a los alumnos desarrollar habilidades para un diseño de programas correcto y eficiente desde un principio. Además, se introduce a los alumnos los problemas que se estudian en el área de Complejidad Computacional.</p> <p>Se requiere para el cursado un buen desarrollo de prácticas de programación, un conocimiento detallado de las principales estructuras de datos, un manejo inicial de las herramientas matemáticas básicas para análisis de algoritmos (como las notaciones asintóticas y elementos de probabilidades) y un conocimiento de los distintos modelos de computabilidad y de los conceptos de problemas computables y semi-computables.</p> <p>El curso comienza con un repaso y profundización del conocimiento y aplicación de las herramientas básicas. Se profundizan e introducen además algunas estructuras de datos que luego van a ser utilizadas a lo largo de la materia. A continuación, se presentan numerosos casos de estudio de análisis de algoritmos, clasificados de acuerdo a la estrategia de resolución de problemas aplicada en cada caso. En cada uno de estos casos de estudio se analiza la aplicación de la estrategia, se estudia el tiempo y el espacio requerido por la ejecución del algoritmo, se presenta la correctitud del algoritmo, se analizan peores y mejores casos, y se compara con otros algoritmos existentes para resolver los mismos problemas. Luego, se introducen las técnicas de análisis amortizado de estructuras de datos, y se presentan algunas estructuras para su aplicación. Después se introduce el concepto de algoritmos probabilísticos, mostrando su clasificación y ejemplos simples. Para finalizar, se presenta el área de Complejidad Computacional, se define el concepto de problemas tratables e intratables, comparándolos con los conceptos análogos de computabilidad, y se discute el problema de <math>P?=NP</math>.</p>					

<b>UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA</b>		2 5
<b>DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN</b>		
<b>ALGORITMOS Y COMPLEJIDAD</b>	<b>CÓDIGO: 5523</b>	
	<b>ÁREA N°: II</b>	

### METODOLOGÍA DE ENSEÑANZA

Las clases teóricas son conducidas por el profesor que presenta los conceptos, las herramientas y las técnicas que conforman los contenidos de la materia. Estas técnicas se ilustran con diversos casos de estudio que emplean diferentes estrategias de solución de problemas. Se promueve la participación de los alumnos con preguntas que permiten retomar y reforzar temas desarrollados en las asignaturas anteriores, fundamentalmente en las materias del área de Programación.

En las clases prácticas el rol de los alumnos es más activo y los docentes brindan pautas que orientan la resolución de los ejercicios y problemas propuestos en los trabajos prácticos y proyectos. Los ejercicios y problemas han sido seleccionados con el objetivo de que, al resolverlos, los alumnos entiendan y apliquen las técnicas herramientas propuestas para el diseño de programas, conozca los recursos necesarios para ejecutar sus algoritmos, sepan dónde buscar algoritmos y librerías para problemas específicos, y diferencien los problemas tratables e intratables.

### MECANISMO DE EVALUACIÓN

La evaluación del cursado se realiza mediante dos parciales, con un recuperatorio final, y dos o tres proyectos. Los parciales se corrigen con nota A-B-C (tres niveles de aprobación) y D-E (dos niveles de desaprobados). Los recuperatorios y los proyectos se corrigen como aprobado/desaprobado, existiendo para los proyectos posibilidad de una re-entrega con correcciones.

Para cursar la materia se requiere aprobar los parciales o sus recuperatorios, y los proyectos. Los alumnos que en los dos parciales obtienen notas A o B, tienen posibilidad de promocionar la materia rindiendo un tercer parcial con los contenidos que no están incluidos en los dos parciales por cuestiones de tiempo.

Los alumnos que cursan la materia pero no acceden a la promoción rinden un examen final escrito donde se evalúan conocimientos teórico-prácticos. El final libre incluye un proyecto, un examen práctico, y un examen teórico-práctico (asimilable a un final regular).

La evaluación de los alumnos en la parte algorítmica se realiza a lo largo de varios ejes: el diseño de algoritmos correctos y eficiente; la aplicación de las técnicas de análisis de algoritmos; la evaluación de la aplicación de estructuras de datos en problemas concretos; y la comprensión de los distintos resultados que proveen los análisis teóricos y empíricos de algoritmos (para esto es fundamental los proyectos que se realizan en la práctica). La evaluación en la parte de complejidad computacional se realiza mediante la comprensión y manejo de los conceptos introducidos, y la diferenciación entre problemas tratables e intratables.



UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA		3 5
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
ALGORITMOS Y COMPLEJIDAD	CÓDIGO: 5523	
	ÁREA N°: II	

### PROGRAMA SINTÉTICO

1. Análisis de Algoritmos: análisis asintótico, comportamiento en el mejor caso, caso promedio y peor caso. Notación  $O()$ . Balance entre tiempo y espacio en los algoritmos.
2. Técnicas y Herramientas para el Análisis de Algoritmos. Estrategias de Diseño de Algoritmos.
3. Algoritmos “greedy”.
4. Algoritmos “dividir y conquistar”.
5. Programación dinámica.
6. Algoritmos sobre Grafos.
7. Análisis Amortizado de Estructuras de Datos.
8. Algoritmos Probabilísticos.
9. Introducción a la Complejidad Computacional. Análisis de complejidad de algoritmos.
10. Clases de Complejidad Computacional. Problema  $P=NP$ . Problemas tratables e intratables.

### PROGRAMA ANALÍTICO

**Capítulo 1 – Introducción:** Conceptos preliminares: Problemas, Algoritmos, Programas. Instancias de un problema. Tamaño de una instancia. Problemas de la Algoritmia: correctitud, eficiencia, aproximación. Reducción entre problemas. Tiempo y espacio de un algoritmo. Tipos de análisis: peor caso, caso promedio, análisis probabilístico, mejor caso. Ejemplos simples de Análisis de Algoritmos.

**Capítulo 2 – Técnicas y Herramientas para el Análisis de Algoritmos -Estrategias de Diseño de Algoritmos:** Técnicas de demostración. Inducción como herramienta de prueba. Inducción constructiva. Repaso de conceptos de lógica, teoría de conjuntos, álgebra, probabilidad y combinatoria. Notación Asintótica: orden, omega y orden exacto. Notación asintótica condicional. Notación asintótica en varios parámetros. Propiedades. Métodos de resolución de Recurrencias: ecuación característica, teorema maestro y cambio de variables. Introducción al Análisis de Algoritmos: tipo de análisis de acuerdo a la estructura de control. Uso de un “barómetro”. Repaso de estructuras de datos: Heaps y Conjuntos Disjuntos.

**Capítulo 3 - Algoritmos “greedy”:** Características generales. Aplicaciones. Esquema general. Pruebas de correctitud. Ejemplos. Problema de los árboles de cubrimientos minimales: algoritmos de Kruskal y de Prim. Posibles implementaciones. Problema de la distancia mínima con origen único: algoritmo de Dijkstra. Problema de la mochila. Problema del viajante. Problema del scheduling de procesos. Demostración de correctitud y análisis de eficiencia.

**Capítulo 4 - Algoritmos “dividir y conquistar”:** Características generales. Aplicaciones. Esquema general. Pruebas de correctitud. Determinación del umbral. Ejemplos. Multiplicación de Enteros Grandes. Búsqueda Binaria. Algoritmos de Ordenamiento:

UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA		4 5
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
ALGORITMOS Y COMPLEJIDAD	CÓDIGO: 5523	
	ÁREA N°: II	

Mergesort, Quicksort. Problema de selección del elemento mediano: algoritmo lineal. Operaciones sobre Matrices: multiplicación de matrices: algoritmo de Strassen, descomposición LUP, matrices inversas. Transformadas de Fourier: representación de polinomios, transformadas de Fourier, implementación eficiente, Problema del par de puntos más cercanos. Problema del cálculo de la exponenciación modular. Aplicación en criptografía. Introducción a la criptografía.

**Capítulo 5 – Programación dinámica:** Características generales. Aplicaciones. Esquema general. Determinación de la recurrencia. Pruebas de correctitud. Ejemplos simples: números de Fibonacci, coeficientes binomiales, probabilidad de ganar una serie de juegos. Ejemplos avanzados. Problema de la mochila. Problema del cambio. Problema de la distancia mínima entre todos los pares de nodos: algoritmo de Floyd. Problema de la multiplicación óptima de matrices. Problema de la triangularización de polígonos. Problema de la subsecuencia más larga. Problema del viajante.

**Capítulo 6 - Algoritmos de Grafos:** Representación de grafos. Algoritmos de recorrido en profundidad y por niveles. Datos generados. Análisis de eficiencia. Aplicaciones. Orden topológico. Componentes fuertemente conexos. Puntos de Articulación. Camino de Euler. Flujo máximo. Demostraciones de correctitud.

**Capítulo 7 - Análisis Amortizado de Estructuras de Datos:** Principios. Definición. Aplicaciones. Método de la función potencial. Ejemplos simples. Definición y análisis amortizado de las estructuras de datos Tablas Dinámicas. Colas Binomiales. Heaps de Fibonacci.

**Capítulo 8 - Algoritmos Probabilísticos:** Generadores de números aleatorios. Clasificación de los Algoritmos Probabilísticos: algoritmo numéricos, algoritmos de Monte Carlo, algoritmos de Las Vegas. Algunos algoritmos probabilísticos simples.

**Capítulo 9 - Introducción a la Complejidad Computacional:** Problemas. Problemas de decisión. Ejemplos. Modelos de Computación. Clases de problemas. Herramientas para la comparación de problemas. Reducción entre problemas Medidas de complejidad Revisión de conceptos de computabilidad.

**Capítulo 10 – Clases de Complejidad Computacional:** Clases P, NP, y NP-completos. Teorema de Cook/Levin. Problema  $P=NP$ . Clases de complejidad en espacio. Otras clases de complejidad computacional. Algoritmos de aproximación.

## BIBLIOGRAFÍA

### Bibliografía Básica

- 1.- "Introduction to Algorithms", 3<sup>rd</sup> edition. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. McGraw Hill-MIT Press 2009.-
- 2.- "P, NP and NP Completeness" O. Goldreich. Cambridge. 2010.



UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA		5 / 5
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
ALGORITMOS Y COMPLEJIDAD	CÓDIGO: 5523 ÁREA N°: II	

**Bibliografía Adicional**

- 1.- "Computational Complexity: A Modern Approach" S. Arora, B. Barak. Cambridge. 2009.
- 2.- "An Introduction to the Analysis of Algorithms", 2<sup>nd</sup> edition. R. Sedgwick, P. Flajolet. Addison Wesley. 2013.
- 3.- "The Design and Analysis of Computer Algorithms." A. V. Aho, J. E. Hopcroft y J. D. Ullman. AddisonWesley. 1983.-
- 4.- "Algorithmics: the Spirit of Computing.", 2<sup>nd</sup> edition. D. Harel. Addison Wesley. 2004.-
- 5.- "Fundamentals of Computer Algorithms." E. Horowitz y S. Sahni. Computer Science Press. 1978.-
- 6.- "The Algorithm Design Manual", 2<sup>nd</sup> edition. S. Skiena. Springer. 2008.-
- 7.- "Fundamentals of Algorithmics". Gilles Brassard, Paul Bratley. Prentice Hall. 1996.-
- 8.- "Introduction to the Theory of Complexity" D. Bovet, P.Crescenzi. Prentice Hall. 1993.-
- 9.- "Analysis of Algorithms and Data Structures." L. Banachowski, A. Kreczmar y W. Rytter. Addison Wesley. 1991.-
- 10.- "Computational Complexity" C. Papadimitriou. Addison-Wesley. 1994.-
- 11.- "Data Structures and Algorithms Analysis" G. Weiss. Benjamin Cummings. 1995.-
- 12.- "Computers and Intractability" M. Garey, D. Johnson. Freeman. 1979.
- 13.- "Combinatorial Optimization: Algorithms and Complexity". C. Papadimitriou, K. Steiglitz. Dover. 1998.-
- 14.- "Introduction to the Design and Analysis of Algorithms". 2<sup>nd</sup> edition. A. Levitin. AddisonWesley. 2006.-
- 15.- "Computational Complexity: A Modern Approach". Sanjeev Arora, Boaz Barak. Cambridge University Press. 2009.-
- 16.- "Complexity: A Guided Tour". Melanie Mitchell. Oxford University Press. 2009.-
- 17.- "Applied Numerical Methods with MATLAB". S. Chapra. McGraw Hill. 2011.-
- 18.- "Introduction to the Theory of Computation", 2<sup>nd</sup> edition. Michael Sipser. PWS Publishing Company. 2005.-

AÑO		FIRMA PROFESOR RESPONSABLE	
2018			
<b>VISADO</b>			
COORDINADOR ÁREA	SECRETARIO ACADÉMICO	DIRECTOR DEPARTAMENTO	
	 Dr. DIEGO C. MARTINEZ SECRETARIO ACADÉMICO DPTO. DE CS. E ING. DE LA COMP. UNIVERSIDAD NACIONAL DEL SUR	 Dr. MARCELO A. FALAPPA DIRECTOR DECANO DPTO. DE CS. E ING. DE LA COMP. UNIVERSIDAD NACIONAL DEL SUR	

UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA		1 5
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
ALGORITMOS Y COMPLEJIDAD	CÓDIGO: 5523 ÁREA N°: II	

**CARRERAS**  
Licenciatura en Ciencias de la Computación

**PROFESOR RESPONSABLE:**  
Dr. Pablo R. Fillottrani – Profesor Asociado con Dedicación Exclusiva

<b>CARGA HORARIA</b>	<b>Teoría</b> 64	<b>Práctica</b> 50	<b>Laboratorio</b> 14	<b>CANTIDAD DE SEMANAS</b>	16
----------------------	---------------------	-----------------------	--------------------------	----------------------------	----

**CORRELATIVAS**

PARA CURSAR LA MATERIA		PARA APROBAR LA MATERIA	
APROBADAS	CURSADAS	APROBADAS	CURSADAS
Tecnología de Programación	Lógica para Cs. de la Computación.  Métodos de Computación Científica.	Lógica para Cs. de la Computación.  Métodos de Computación Científica.	

**DESCRIPCIÓN**

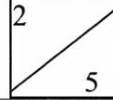
El objetivo específico de esta materia es afianzar el uso de técnicas de análisis de algoritmos que permitan a los alumnos desarrollar habilidades para un diseño de programas correcto y eficiente desde un principio. Además, se introduce a los alumnos los problemas que se estudian en el área de Complejidad Computacional.

Se requiere para el cursado un buen desarrollo de prácticas de programación, un conocimiento detallado de las principales estructuras de datos, un manejo inicial de las herramientas matemáticas básicas para análisis de algoritmos (como las notaciones asintóticas y elementos de probabilidades) y un conocimiento de los distintos modelos de computabilidad y de los conceptos de problemas computables y semi-computables.

El curso comienza con un repaso y profundización del conocimiento y aplicación de las herramientas básicas. Se profundizan e introducen además algunas estructuras de datos que luego van a ser utilizadas a lo largo de la materia. A continuación, se presentan numerosos casos de estudio de análisis de algoritmos, clasificados de acuerdo a la estrategia de resolución de problemas aplicada en cada caso. En cada uno de estos casos de estudio se analiza la aplicación de la estrategia, se estudia el tiempo y el espacio requerido por la ejecución del algoritmo, se presenta la correctitud del algoritmo, se analizan peores y mejores casos, y se compara con otros algoritmos existentes para resolver los mismos problemas. Luego, se introducen las técnicas de análisis amortizado de estructuras de datos, y se presentan algunas estructuras para su aplicación. Después se introduce el concepto de algoritmos probabilísticos, mostrando su clasificación y ejemplos simples. Para finalizar, se presenta el área de Complejidad Computacional, se define el concepto de problemas tratables e intratables, comparándolos con los conceptos análogos de computabilidad, y se discute el problema de  $P?=NP$ .



**UNIVERSIDAD NACIONAL DEL SUR  
BAHÍA BLANCA**



**DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN**

**ALGORITMOS Y COMPLEJIDAD**

**CÓDIGO: 5523**

**ÁREA N°: II**

### **METODOLOGÍA DE ENSEÑANZA**

Las clases teóricas son conducidas por el profesor que presenta los conceptos, las herramientas y las técnicas que conforman los contenidos de la materia. Estas técnicas se ilustran con diversos casos de estudio que emplean diferentes estrategias de solución de problemas. Se promueve la participación de los alumnos con preguntas que permiten retomar y reforzar temas desarrollados en las asignaturas anteriores, fundamentalmente en las materias del área de Programación.

En las clases prácticas el rol de los alumnos es más activo y los docentes brindan pautas que orientan la resolución de los ejercicios y problemas propuestos en los trabajos prácticos y proyectos. Los ejercicios y problemas han sido seleccionados con el objetivo de que, al resolverlos, los alumnos entiendan y apliquen las técnicas herramientas propuestas para el diseño de programas, conozca los recursos necesarios para ejecutar sus algoritmos, sepan dónde buscar algoritmos y librerías para problemas específicos, y diferencien los problemas tratables e intratables.

### **MECANISMO DE EVALUACIÓN**

La evaluación del cursado se realiza mediante dos parciales, con un recuperatorio final, y dos o tres proyectos. Los parciales se corrigen con nota A-B-C (tres niveles de aprobación) y D-E (dos niveles de desaprobados). Los recuperatorios y los proyectos se corrigen como aprobado/desaprobado, existiendo para los proyectos posibilidad de una re-entrega con correcciones.

Para cursar la materia se requiere aprobar los parciales o sus recuperatorios, y los proyectos. Los alumnos que en los dos parciales obtienen notas A o B, tienen posibilidad de promocionar la materia rindiendo un tercer parcial con los contenidos que no están incluidos en los dos parciales por cuestiones de tiempo.

Los alumnos que cursan la materia pero no acceden a la promoción rinden un examen final escrito donde se evalúan conocimientos teórico-prácticos. El final libre incluye un proyecto, un examen práctico, y un examen teórico-práctico (asimilable a un final regular).

La evaluación de los alumnos en la parte algorítmica se realiza a lo largo de varios ejes: el diseño de algoritmos correctos y eficiente; la aplicación de las técnicas de análisis de algoritmos; la evaluación de la aplicación de estructuras de datos en problemas concretos; y la comprensión de los distintos resultados que proveen los análisis teóricos y empíricos de algoritmos (para esto es fundamental los proyectos que se realizan en la práctica). La evaluación en la parte de complejidad computacional se realiza mediante la comprensión y manejo de los conceptos introducidos, y la diferenciación entre problemas tratables e intratables.

### PROGRAMA SINTÉTICO

1. Análisis de Algoritmos: análisis asintótico, comportamiento en el mejor caso, caso promedio y peor caso. Notación  $O()$ . Balance entre tiempo y espacio en los algoritmos.
2. Técnicas y Herramientas para el Análisis de Algoritmos. Estrategias de Diseño de Algoritmos.
3. Algoritmos "greedy".
4. Algoritmos "dividir y conquistar".
5. Programación dinámica.
6. Algoritmos sobre Grafos.
7. Análisis Amortizado de Estructuras de Datos.
8. Algoritmos Probabilísticos.
9. Introducción a la Complejidad Computacional. Análisis de complejidad de algoritmos.
10. Clases de Complejidad Computacional. Problema  $P=NP$ . Problemas tratables e intratables.

### PROGRAMA ANALÍTICO

**Capítulo 1 – Introducción:** Conceptos preliminares: Problemas, Algoritmos, Programas. Instancias de un problema. Tamaño de una instancia. Problemas de la Algoritmia: correctitud, eficiencia, aproximación. Reducción entre problemas. Tiempo y espacio de un algoritmo. Tipos de análisis: peor caso, caso promedio, análisis probabilístico, mejor caso. Ejemplos simples de Análisis de Algoritmos.

**Capítulo 2 – Técnicas y Herramientas para el Análisis de Algoritmos -Estrategias de Diseño de Algoritmos:** Técnicas de demostración. Inducción como herramienta de prueba. Inducción constructiva. Repaso de conceptos de lógica, teoría de conjuntos, álgebra, probabilidad y combinatoria. Notación Asintótica: orden, omega y orden exacto. Notación asintótica condicional. Notación asintótica en varios parámetros. Propiedades. Métodos de resolución de Recurrencias: ecuación característica, teorema maestro y cambio de variables. Introducción al Análisis de Algoritmos: tipo de análisis de acuerdo a la estructura de control. Uso de un "barómetro". Repaso de estructuras de datos: Heaps y Conjuntos Disjuntos.

**Capítulo 3 - Algoritmos "greedy":** Características generales. Aplicaciones. Esquema general. Pruebas de correctitud. Ejemplos. Problema de los árboles de cubrimientos minimales: algoritmos de Kruskal y de Prim. Posibles implementaciones. Problema de la distancia mínima con origen único: algoritmo de Dijkstra. Problema de la mochila. Problema del viajante. Problema del scheduling de procesos. Demostración de correctitud y análisis de eficiencia.

**Capítulo 4 - Algoritmos "dividir y conquistar":** Características generales. Aplicaciones. Esquema general. Pruebas de correctitud. Determinación del umbral. Ejemplos. Multiplicación de Enteros Grandes. Búsqueda Binaria. Algoritmos de Ordenamiento:



Mergesort, Quicksort. Problema de selección del elemento mediano: algoritmo lineal. Operaciones sobre Matrices: multiplicación de matrices: algoritmo de Strassen, descomposición LUP, matrices inversas. Transformadas de Fourier: representación de polinomios, transformadas de Fourier, implementación eficiente, Problema del par de puntos más cercanos. Problema del cálculo de la exponenciación modular. Aplicación en criptografía. Introducción a la criptografía.

**Capítulo 5 – Programación dinámica:** Características generales. Aplicaciones. Esquema general. Determinación de la recurrencia. Pruebas de correctitud. Ejemplos simples: números de Fibonacci, coeficientes binomiales, probabilidad de ganar una serie de juegos. Ejemplos avanzados. Problema de la mochila. Problema del cambio. Problema de la distancia mínima entre todos los pares de nodos: algoritmo de Floyd. Problema de la multiplicación óptima de matrices. Problema de la triangularización de polígonos. Problema de la subsecuencia más larga. Problema del viajante.

**Capítulo 6 - Algoritmos de Grafos:** Representación de grafos. Algoritmos de recorrido en profundidad y por niveles. Datos generados. Análisis de eficiencia. Aplicaciones. Orden topológico. Componentes fuertemente conexos. Puntos de Articulación. Camino de Euler. Flujo máximo. Demostraciones de correctitud.

**Capítulo 7 - Análisis Amortizado de Estructuras de Datos:** Principios. Definición. Aplicaciones. Método de la función potencial. Ejemplos simples. Definición y análisis amortizado de las estructuras de datos Tablas Dinámicas. Colas Binomiales. Heaps de Fibonacci.

**Capítulo 8 - Algoritmos Probabilísticos:** Generadores de números aleatorios. Clasificación de los Algoritmos Probabilísticos: algoritmo numéricos, algoritmos de Monte Carlo, algoritmos de Las Vegas. Algunos algoritmos probabilísticos simples.

**Capítulo 9 - Introducción a la Complejidad Computacional:** Problemas. Problemas de decisión. Ejemplos. Modelos de Computación. Clases de problemas. Herramientas para la comparación de problemas. Reducción entre problemas Medidas de complejidad Revisión de conceptos de computabilidad.

**Capítulo 10 – Clases de Complejidad Computacional:** Clases P, NP, y NP-completos. Teorema de Cook/Levin. Problema  $P \stackrel{?}{=} NP$ . Clases de complejidad en espacio. Otras clases de complejidad computacional. Algoritmos de aproximación.

## BIBLIOGRAFÍA





### Bibliografía Básica

- 1.- "Introduction to Algorithms", 3<sup>rd</sup> edition. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. McGraw Hill-MIT Press 2009.-
- 2.- "P, NP and NP Completeness" O. Goldreich. Cambridge. 2010.

UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA		5 5
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
ALGORITMOS Y COMPLEJIDAD	CÓDIGO: 5523 ÁREA Nº: II	

### Bibliografía Adicional

- 1.- "Computational Complexity: A Modern Approach" S. Arora, B. Barak. Cambridge. 2009.
- 2.- "An Introduction to the Analysis of Algorithms", 2<sup>nd</sup> edition. R. Sedgewick, P. Flajolet. Addison Wesley. 2013.
- 3.- "The Design and Analysis of Computer Algorithms." A. V. Aho, J. E. Hopcroft y J. D. Ullman. AddisonWesley. 1983.-
- 4.- "Algorithmics: the Spirit of Computing.", 2<sup>nd</sup> edition. D. Harel. Addison Wesley. 2004.-
- 5.- "Fundamentals of Computer Algorithms." E. Horowitz y S. Sahni. Computer Science Press. 1978.-
- 6.- "The Algorithm Design Manual", 2<sup>nd</sup> edition. S. Skiena. Springer. 2008.-
- 7.- "Fundamentals of Algorithmics". Gilles Brassard, Paul Bratley. Prentice Hall. 1996.-
- 8.- "Introduction to the Theory of Complexity" D. Bovet, P.Crescenzi. Prentice Hall. 1993.-
- 9.- "Analysis of Algorithms and Data Structures." L. Banachowski, A. Kreczmar y W. Rytter. Addison Wesley. 1991.-
- 10.- "Computational Complexity" C. Papadimitriou. Addison-Wesley. 1994.-
- 11.- "Data Structures and Algorithms Analysis" G. Weiss. Benjamin Cummings. 1995.-
- 12.- "Computers and Intractability" M. Garey, D. Johnson. Freeman. 1979.
- 13.- "Combinatorial Optimization: Algorithms and Complexity". C. Papadimitriou, K. Steiglitz. Dover. 1998.-
- 14.- "Introduction to the Design and Analysis of Algorithms". 2<sup>nd</sup> edition. A. Levitin. AddisonWesley. 2006.-
- 15.- "Computational Complexity: A Modern Approach". Sanjeev Arora, Boaz Barak. Cambridge University Press. 2009.-
- 16.- "Complexity: A Guided Tour". Melanie Mitchell. Oxford University Press. 2009.-
- 17.- "Applied Numerical Methods with MATLAB". S. Chapra. McGraw Hill. 2011.-
- 18.- "Introduction to the Theory of Computation", 2<sup>nd</sup> edition. Michael Sipser. PWS Publishing Company. 2005.-

AÑO		FIRMA PROFESOR RESPONSABLE	
2018			
<b>VISADO</b>			
COORDINADOR ÁREA	SECRETARIO ACADÉMICO	DIRECTOR DEPARTAMENTO	
	 Dr. DIEGO C. MARTINEZ SECRETARIO ACADÉMICO DPTO. DE CS. E ING. DE LA COMP. UNIVERSIDAD NACIONAL DEL SUR	 Dr. MARCELO A. FALAPPA DIRECTOR DECANO DPTO. DE CS. E ING. DE LA COMP. UNIVERSIDAD NACIONAL DEL SUR	