

Taller de Tecnologías de Producción de Software

TÉCNICAS Y ESTRATEGIAS PARA LA RESOLUCIÓN DE PROBLEMAS

Facultad de Informática - UNLP

Año 2025

Técnicas y Estrategias para la Resolución de Problemas

- Docentes
- Objetivos
- Organización
- Pautas generales a tener en cuenta al resolver un problema
- Competencias de Programación
- Jueces on-line
- Referencias

Docentes

Prof. Alejandra Schiavoni

Auxiliares:

JTP: Matías Fluxa

AD: Juan Ignacio Vergara

Objetivos

- Reforzar y ampliar los conocimientos con el fin de perfeccionar las habilidades en la resolución de problemas complejos
- Plantear problemas basados en situaciones de la vida real de diferentes niveles de complejidad
- Introducir a los alumnos en un análisis minucioso de cada problema a resolver

Organización del curso

- Consistirá en explicaciones de temas teóricos y su aplicación en problemas concretos
- Planteo de problemas de dificultad gradual para que los alumnos resuelvan e implementen en clase y en su casa
- Discusión de posibles soluciones propuestas por los alumnos
- Testeo de las distintas soluciones con diferentes casos de prueba

Técnicas para resolver problemas

Utilidad en la vida laboral

- Facilidad en la resolución de problemas complejos.
- Optimización en la era de la eficiencia.
- Habilidad para analizar un problema.

Pautas generales a tener en cuenta al resolver un problema

- Análisis del problema
- Pistas básicas de programación
- Diseño de casos de prueba

Análisis del problema

- Es importante leer e interpretar las especificaciones del problema apropiadamente
- *Nunca* asumir hechos o datos que no están establecidos explícitamente
- Una vez comprendido el problema, escribir un algoritmo que lo resuelva

Categorías de los problemas

En general los problemas a resolver pueden clasificarse de acuerdo a la técnica necesaria para resolverlos. Algunas de las categorías son:

- Estructuras de datos avanzadas
- Algoritmos sobre strings
- Geometría computacional
- Problemas de carácter matemático
- Backtracking
- Divide y Conquer
- Programación dinámica
- Greedy algorithms
- Teoría de grafos
- Ad-Hoc

Pistas básicas de programación

- Agregar comentarios al comienzo (funciones, programas, etc.)
- Uso de variables:
 - Documentar cada variable
 - No usar nombres muy cortos y similares
 - En los loops usar: {I, K, M} en lugar de {i, j, k}
- Usar constantes simbólicas
- Usar tipos enumerativos con algún motivo
- Usar subrutinas para evitar código redundante
- Usar sentencias de debugging significativas

Diseño de casos de prueba

- Deben incluir el ejemplo dado, es trivial y se tiene la respuesta
- Deben incluir casos límite
- Para múltiples casos de entrada, probar usando dos casos idénticos consecutivamente
- Incrementar el tamaño de la entrada
- Armar casos de prueba tramposos
- No asumir que la entrada respeta un formato
- Construir casos de prueba aleatorios

Competencias de Programación

- ¿Qué son?
- ¿Cuáles son las más conocidas?
- Competencia Internacional de Programación de ACM

¿Qué es una Competencia de Programación?

- En ella los participantes deben resolver algoritmos complejos en un límite de tiempo
- Constan de varias rondas, primero a nivel nacional o regional y al finalizar a nivel mundial
- La solución de los problemas puede realizarse en distintos lenguajes
- En general la participación es en equipos
- El objetivo es promover la creatividad, la innovación y les permite a los concursantes testear su habilidad para actuar bajo presión

¿Cuáles son las más conocidas?

- ACM–ICPC: ACM International Collegiate Programming Contest

<https://icpc.baylor.edu/>

- International Olympiad in Informatics (IOI)
(Para nivel escolar)

<https://ioinformatics.org/>

Competencia Internacional de Programación de ACM



- Es la mayor y más antigua a nivel internacional
- Se inicia con competencias locales y regionales en todo el mundo, de las cuales salen los equipos que participan cada año en las ACM-ICPC World Finals

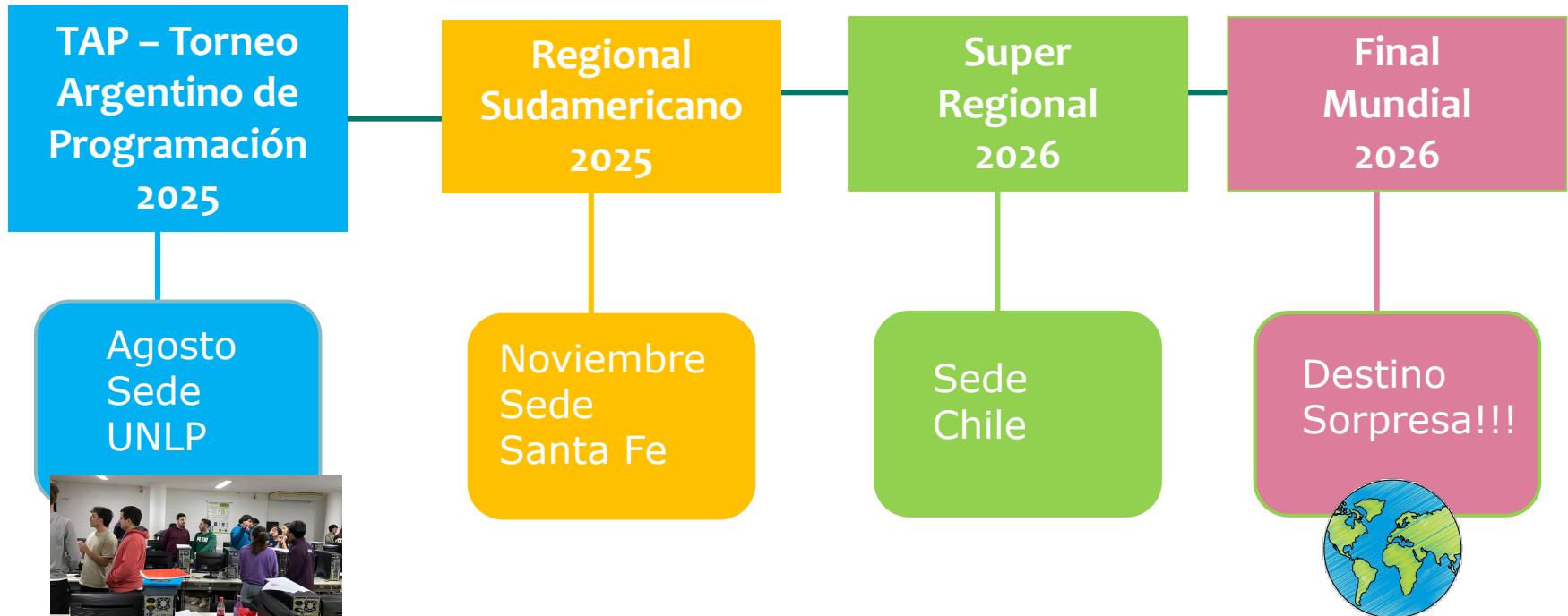
Competencia Internacional de Programación de ACM

Instancias 2024 - 2025



Competencia Internacional de Programación de ACM

Instancias 2025 - 2026



TAP: Torneo Argentino de Programación



Torneo Argentino de Programación

- Se enmarca dentro de la competencia ACM-ICPC.
- Sedes en todo el país (2025): Bariloche, Buenos Aires, Chilecito, Córdoba, Jujuy, La Plata, Nueva Orán, Resistencia, Rosario, Río Cuarto, Santa Fe y Tucumán.



Horas



Lenguajes
Python, C/C++, Java,
Kotlin



Integrantes

TAP: Torneo Argentino de Programación



Jueces on-line

- ¿Qué son?
- Ejemplos
- Formato de los problemas
- Respuestas posibles

¿Qué son los Jueces on-line?

- Es posible tener una evaluación inmediata de las soluciones a través de robots que actúan como jueces
- Esta forma de testear los algoritmos incentiva el desafío y mejora las habilidades de programación
- Los jueces on-line corrigen los programas de la misma forma que lo hacen los jueces en la competencia de ACM.
- Se crea una cuenta para mandar los algoritmos a ser evaluados
- Las soluciones pueden estar en distintos lenguajes.

Ejemplos de Jueces on-line

Existen varios jueces on-line, entre los más importantes están:

- **UVa Online Judge:** Gran diversidad de problemas de competencias locales, es el primer juez on line. <http://uva.onlinejudge.org/>
- **Sphere Online Judge (SPOJ):** Juez con una gran cantidad de problemas de todo tipo. <http://www.spoj.com/>
- **Codeforces:** <https://codeforces.com/>

Resulta útil acceder a Virtual Judge <https://vjudge.net/> , que no es un juez online, pero permite simular el envío a otros jueces (incluye un gran conjunto de jueces)

Formato de los problemas

Todos los problemas en las competencias tienen las siguientes características:

- En general están escritos en inglés.
- El planteo de los problemas se hace en forma de “historia”, obligando a realizar una abstracción del mismo.
- El enunciado se encuentra dividido en: planteo, descripción de la entrada y descripción de la salida.
- No puede asumirse nada acerca de lo que no está explicitado en el enunciado.

Respuestas posibles

Las respuestas de estos jueces en algunos casos, dan pocas pistas (feedback) sobre los errores del programa

Los posibles resultados son:

- **Accepted:** La solución ingresada es válida.
- **Wrong Answer:** El programa compiló y corrió correctamente pero la salida no es la esperada.
- **Presentation Error:** El programa compiló, corrió y las respuestas son las correctas pero no sigue el formato establecido.
- **Compilation Error:** El código fuente no compila.
- **Runtime Error:** El programa no terminó de forma correcta. (Segmentation Fault, NullPointerException, etc)
- **Time Limit Exceeded:** El programa corrió por más tiempo del permitido. (Loop infinito o la solución no es la óptima)
- **Memory Limit Exceeded:** El programa usó más memoria de la permitida.

Referencias

- **“Programming Challenges: The Programming Contest Training Manual”**. Steven S. Skiena, Miguel A. Revilla. Springer-Verlag New York, Inc., 2003 ISBN 0-387-00163-8
- **“Competitive Programming 3”**. Steven Halim, Felix Halim. Handbook for ACM ICPC and IOI Contestants 2013
- **“Art of Programming Contest”**. Ahmed Shamsul Arefin, Publisher Gyankosh Prokashoni, 2006, ISBN : 984-32-3382-4
- **“Programming Pearls”**. Second Edition. Jon Bentley. Addison-Wesley, Inc., 2000. ISBN 0-201-65788-0.
- **“Algorithms”**. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani. McGraw-Hill Science/Engineering/Math; 1 edition (September 13, 2006). ISBN-13: 978-0073523408