



TÉCNICAS Y ESTRATEGIAS PARA LA RESOLUCIÓN DE PROBLEMAS

Facultad de Informática - UNLP

Año 2021



Técnicas y Estrategias para la Resolución de Problemas

2

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

Docentes

3

Prof. Alejandra Schiavoni

Auxiliares:

Andrés Cimadamore

Duilio Ray

Objetivos

4

- 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

5

- 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

6

- 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

7

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

Análisis del problema

8

- 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

9

En general los problemas a resolver pueden clasificarse de acuerdo a la técnica necesaria para resolverlos. 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

10

- Agregar comentarios al comienzo (funciones, programas, etc.)
- Uso de variables:
 - Documentar cada variable
 - No usar nombres muy cortos y similares
 - En los loops usar: $\{l, K, M\}$ 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

11

- 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

12

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

¿Qué es una Competencia de Programación?

13

- 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?

14

- ACM–ICPC: ACM International Collegiate Programming Contest

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

- Google Code Jam

<http://code.google.com/codejam/>

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

<http://ioinformatics.org/index.shtml>

Competencia Internacional de Programación de ACM

15



- Es la mayor y más antigua a nivel internacional
- Se inicia con competencias regionales en todo el mundo, de las cuales salen los equipos que participan cada año en las ACM-ICPC World Finals
- La competencia regional sudamericana se lleva a cabo simultáneamente en Chile, Colombia, Venezuela, Perú, Brasil, Argentina y Bolivia. Una de las sedes es en la UBA.

TAP: Torneo Argentino de Programación

16



- Se enmarca dentro de la competencia ACM-ICPC.
- Las instituciones que así lo deseen (no es obligatorio) pueden usar el torneo para seleccionar sus equipos para la Competencia Regional Latinoamericana.
- Sedes en todo el país (2019): Bahía Blanca (UNS), Buenos Aires (UBA), Chilecito (UNdeC), Córdoba (UNC), Jujuy (UNJu), La Plata (UNLP), La Rioja (UNLaR), Rosario (UNR), Orán (UNS) y Tucumán (UNT)
- Es una competencia por equipos de 3 estudiantes de la misma institución de educación superior de Argentina. La competencia tiene 5 horas de duración, en las que cada equipo deberá resolver un conjunto de problemas algorítmicos, creando un programa que solucione cada problema.

TAP: Torneo Argentino de Programación



17



Jueces on-line

18

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

¿Qué son los Jueces on-line?

19

- 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

20

Existen varios jueces on-line, entre los más importantes y que usamos en la cursada son:

- **UVa Online Judge:** Gran diversidad de problemas de competencias locales.
<http://uva.onlinejudge.org/>
- **Sphere Online Judge (SPOJ):** Juez con una gran cantidad de problemas de todo tipo. <http://www.spoj.com/>
- **Caribbean Online Judge (COJ):** <http://coj.uci.cu/index.xhtml>
- **LiveArchive:** Contiene los problemas de las competencias regionales y finales de la ACM-ICPC. <http://livearchive.onlinejudge.org/>
- **URI Online Judge:**
<https://www.urionlinejudge.com.br/judge/login?redirect=%2Fen%2Flogin>

Lenguajes aceptados por los jueces

21

A continuación se indican los lenguajes aceptados por cada juez:

➤ **LiveArchive**

C

C++

Java

➤ **UVa Online Judge**

C

C++

Java

Python

➤ **Sphere Online Judge (SPOJ)**

C

C++

Java

Python

Pascal

... muchos más

➤ **COJ**

C

C++

Java

Python

PHP

.... varios más

Formato de los problemas

22

Todos los problemas en las competencias tienen las siguientes características:

- 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

23

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 mas memoria de la permitida.

Referencias

24

- **“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