

Olimpiada Informática Argentina  
Categoría “Programación”

## Temario orientativo por nivel

### Aclaraciones:

- Los conocimientos que se listan, son **orientativos** para los alumnos, docentes, y entrenadores, y son **no excluyentes** de otros conocimientos que se podrían incorporar ocasionalmente en los problemas de los Certámenes.
- Estos conocimientos son acumulativos para los distintos niveles (Se considera que los temas de cada nivel incluyen los temas correspondientes a los niveles anteriores).
- El temario está pensado como una orientación general de los temas que **podrían** aparecer en un problema, incluso en los más difíciles. Por lo tanto:
  - **No se espera** que los participantes necesariamente conozcan y dominen **todos** los temas del temario.
  - Se espera que los problemas más sencillos requieran únicamente de los conocimientos más básicos.

# Nivel 1

Teoría: Matemática	Teoría: Ciencias de la computación	Algunos problemas de ejemplo
<p><b>Aritmética y Geometría:</b></p> <ul style="list-style-type: none"> <li>• Enteros, operaciones aritméticas, comparación</li> <li>• Sistemas de numeración, conversión entre ellos</li> <li>• Propiedades básicas de los enteros (signo, paridad, divisibilidad)</li> <li>• Aritmética modular básica: suma, resta, multiplicación</li> <li>• Fracciones, porcentajes</li> <li>• Números primos</li> </ul> <p><b>Lógica básica:</b></p> <ul style="list-style-type: none"> <li>• Entender enunciados matemáticos con “existe” y “para todo”, y su diferencia.</li> <li>• Conectivos lógicos (<b>y</b>, <b>o</b>, <b>no</b>, <b>entonces</b>).</li> <li>• Uso de razonamientos y deducciones lógicas</li> </ul>	<p><b>Programación:</b></p> <ul style="list-style-type: none"> <li>• Sintaxis y semántica básicas de algún lenguaje permitido en OIA</li> <li>• Variables, tipos, expresiones y asignación</li> <li>• Entrada y salida sencilla (desde y hacia archivos y stdin / stdout)</li> <li>• Estructuras de control selectivas (if)</li> <li>• Estructuras de control repetitivas (while / for / repeat)</li> <li>• Funciones (subrutinas) y pasaje de parámetros en ambos sentidos</li> <li>• Descomposición de problemas (pensamiento top-down)</li> </ul> <p><b>Estructuras de datos fundamentales:</b></p> <ul style="list-style-type: none"> <li>• Tipos primitivos (booleanos, enteros con y sin signo, caracteres)</li> <li>• Arreglos unidimensionales</li> <li>• Cadenas y procesamiento básico de cadenas</li> <li>• Variables globales y locales</li> </ul> <p><b>Estrategias algorítmicas:</b></p> <ul style="list-style-type: none"> <li>• Iteración</li> </ul> <p><b>Algoritmos:</b></p> <ul style="list-style-type: none"> <li>• <b>Algoritmos sencillos con enteros:</b> <ul style="list-style-type: none"> <li>• Cambio de base</li> <li>• Test de primalidad (buscar divisores hasta la raíz)</li> <li>• Factorización (buscando divisores)</li> </ul> </li> <li>• <b>Manipulación de arreglos:</b> <ul style="list-style-type: none"> <li>• Llenar</li> <li>• Desplazar</li> <li>• Rotar</li> <li>• Invertir</li> <li>• Mínimo/Máximo</li> <li>• Sumas parciales</li> </ul> </li> <li>• Algoritmos sencillos con cadenas de texto           <ul style="list-style-type: none"> <li>• Búsqueda directa de una subcadena dada.</li> <li>• Invertir una cadena.</li> <li>• Convertir entre mayúsculas y minúsculas.</li> </ul> </li> <li>• Procesamiento y búsqueda secuencial sobre listas de valores (arreglos).</li> </ul>	<ul style="list-style-type: none"> <li>• Construir un arreglo con ciertos elementos especiales de un conjunto (Ej: El máximo, los números primos, los divisibles por 7 y por 11, etc)</li> <li>• Convertir números entre sistemas de numeración</li> <li>• Realizar operaciones con números y escribir sus resultados</li> <li>• Procesar un texto identificando cadenas que contienen ciertas subcadenas definidas</li> <li>• Procesar un texto identificando o corrigiendo palabras de acuerdo a un criterio definido (identificar palíndromos, mayúsculas/minúsculas, letras prohibidas, etc)</li> </ul>

## Nivel 2

Teoría: Matemática	Teoría: Ciencias de la computación	Algunos problemas de ejemplo
<p><b>Aritmética y Geometría:</b></p> <ul style="list-style-type: none"> <li>• Recta, segmento, ángulo, triángulo, rectángulo, cuadrado, circunferencia</li> <li>• Teorema de pitágoras</li> </ul> <p><b>Funciones, relaciones y conjuntos:</b></p> <ul style="list-style-type: none"> <li>• Funciones (inyectivas, sobreyectivas, inversas, composición)</li> <li>• Relaciones (Propiedad de ser simétricas, transitivas)</li> <li>• Orden lexicográfico</li> <li>• Conjuntos (inclusión, complementos, disjuntos)</li> </ul> <p><b>Lógica básica:</b></p> <ul style="list-style-type: none"> <li>• Tablas de verdad</li> </ul> <p><b>Técnicas de demostración:</b></p> <ul style="list-style-type: none"> <li>• Nociones de implicación, recíproco, contrarrecíproco, negación, contradicción</li> <li>• Demostraciones directas, por contraejemplo, por absurdo</li> <li>• Inducción matemática</li> <li>• Definiciones matemáticas recursivas</li> </ul> <p><b>Combinatoria básica:</b></p> <ul style="list-style-type: none"> <li>• Argumentos de conteo (principio de suma y multiplicación, progresiones aritméticas y geométricas, números de Fibonacci)</li> <li>• Factorial, números combinatorios</li> <li>• Principio del palomar</li> </ul> <p><b>Grafos y árboles:</b></p> <ul style="list-style-type: none"> <li>• Árboles y sus propiedades básicas, árboles con raíz</li> <li>• Grafos no dirigidos (grados, camino, ciclo, conectividad)</li> <li>• Grafos dirigidos (grado de entrada, grado de salida, camino y ciclo dirigido)</li> <li>• Árboles generadores</li> <li>• Grafos con pesos, etiquetas, colores</li> <li>• Multigrafos, grafos con bucles</li> </ul>	<p><b>Estructuras de datos fundamentales:</b></p> <ul style="list-style-type: none"> <li>• Arreglos multidimensionales</li> <li>• Registros / Tipos de Datos definidos por el usuario</li> <li>• Estructuras enlazadas (listas, árboles binarios, árboles con raíz, y similares estructuras naturalmente recursivas)</li> <li>• Estrategias de implementación de grafos y árboles</li> </ul> <p><b>Recursión:</b></p> <ul style="list-style-type: none"> <li>• Implementación de funciones matemáticas recursivas</li> </ul> <p><b>Análisis de algoritmos:</b></p> <ul style="list-style-type: none"> <li>• Complejidades estándar (constante, logarítmica, lineal, <math>N \lg N</math>, cuadrática, cúbica, exponencial)</li> <li>• Medición empírica de la eficiencia</li> </ul> <p><b>Estrategias algorítmicas:</b></p> <ul style="list-style-type: none"> <li>• Algoritmos de fuerza bruta / búsqueda exhaustiva</li> <li>• Algoritmos golosos</li> </ul> <p><b>Algoritmos:</b></p> <ul style="list-style-type: none"> <li>• <b>Algoritmos sencillos con enteros:</b> <ul style="list-style-type: none"> <li>• Algoritmo de Euclides</li> <li>• Criba de Eratóstenes</li> <li>• Factorización (con criba)</li> </ul> </li> <li>• Operaciones sencillas con enteros de precisión arbitraria (suma, resta, multiplicación cuadrática)</li> <li>• <b>Manipulación de arreglos:</b> <ul style="list-style-type: none"> <li>• Redimensionar</li> <li>• Histogramas</li> <li>• Bucket / counting sort</li> </ul> </li> <li>• Búsqueda binaria</li> <li>• Quicksort y Quickselect (k-ésimo elemento)</li> <li>• Heapsort y Mergesort</li> <li>• DFS y BFS</li> <li>• Recorrido Inorder, Preorder, Postorder</li> <li>• Aplicaciones de DFS (Orden topológico)</li> <li>• Componentes conexas</li> <li>• Algoritmo de Dijkstra</li> <li>• Árbol generador mínimo (Prim / Kruskal)</li> </ul> <p><b>Estructuras de datos:</b></p> <ul style="list-style-type: none"> <li>• Pilas y colas</li> <li>• Representación de grafos (listas de adyacencia, matriz de adyacencia)</li> <li>• Heap binario</li> <li>• Estructura de Union-Find (Listas de componentes + Arreglo de representantes)</li> </ul>	<ul style="list-style-type: none"> <li>• Ordenar un conjunto de datos de acuerdo a un criterio</li> <li>• Encontrar el camino más corto en un mapa</li> <li>• Dado un mapa, encontrar el máximo número de ciudades que están conectadas entre sí por caminos</li> <li>• En una tabla de <math>N \times M</math> letras, encontrar el conjunto máximo de letras iguales unidas entre sí</li> <li>• Dada una descripción de un sistema de archivos, reportar los directorios que contienen más cantidad de archivos, contando y sin contar subdirectorios, o cuántos son dichos archivos</li> </ul>

## Nivel 3

Teoría: Matemática	Teoría: Ciencias de la computación	Algunos problemas de ejemplo
<p><b>Aritmética y Geometría:</b></p> <ul style="list-style-type: none"> <li>• Punto, vector, coordenadas en el plano</li> <li>• Polígono (vértices, lados, simple, convexo, interior, área)</li> <li>• Distancia euclídea</li> </ul> <p><b>Funciones, relaciones y conjuntos:</b></p> <ul style="list-style-type: none"> <li>• Relaciones (de equivalencia, de orden, de orden total)</li> <li>• Conjuntos (producto cartesiano, conjunto de partes)</li> </ul> <p><b>Combinatoria básica:</b></p> <ul style="list-style-type: none"> <li>• Principio de inclusión/exclusión</li> </ul> <p><b>Grafos y árboles:</b></p> <ul style="list-style-type: none"> <li>• Grafos no dirigidos (ciclo y camino euleriano, ciclo y camino hamiltoniano)</li> <li>• Grafos dirigidos (ciclo y camino euleriano dirigido, ciclo y camino hamiltoniano dirigido)</li> <li>• Grafos bipartitos</li> <li>• Grafos planares (propiedades básicas)</li> </ul>	<p><b>Estructuras de datos fundamentales:</b></p> <ul style="list-style-type: none"> <li>• Uso elemental de números de punto flotante en cálculos estables</li> <li>• Representación de datos en memoria</li> <li>• Memoria Heap vs Stack</li> <li>• Implementación de fracciones para cálculos exactos</li> </ul> <p><b>Análisis de algoritmos:</b></p> <ul style="list-style-type: none"> <li>• Especificación, precondition, poscondition, correctitud, invariantes</li> <li>• Análisis asintótico de complejidad, y notación de <math>O</math> grande.</li> </ul> <p><b>Estrategias algorítmicas:</b></p> <ul style="list-style-type: none"> <li>• Divide y vencerás</li> <li>• Backtracking (recursivo y no recursivo), Branch-and-bound</li> <li>• Programación dinámica</li> </ul> <p><b>Algoritmos:</b></p> <ul style="list-style-type: none"> <li>• Exponenciación eficiente (especialmente con aritmética modular)</li> <li>• Aplicaciones de DFS (encontrar ciclo/camino euleriano)</li> <li>• Algoritmos de Bellman Ford, Floyd-Warshall</li> <li>• Matching máximo bipartito <math>O(VE)</math></li> <li>• Biconectividad en grafos no dirigidos (puentes, puntos de articulación)</li> <li>• Conectividad en grafos dirigidos (componentes fuertemente conexas)</li> </ul> <p><b>Estructuras de datos:</b></p> <ul style="list-style-type: none"> <li>• Árboles binarios balanceados estáticos (Fenwick Tree y Segment Tree)</li> <li>• Árbol binario de búsqueda balanceado (Treap - Splay Tree - AVL u otro)</li> <li>• Árbol binario de búsqueda aumentado</li> <li>• Consulta de LCA en tiempo logarítmico</li> <li>• Estructuras de datos persistentes</li> <li>• Composición de estructuras de datos (Ej: Fenwick Tree en 2 dimensiones)</li> <li>• Tries</li> </ul> <p><b>Algoritmos avanzados:</b></p> <ul style="list-style-type: none"> <li>• Nociones básicas de teoría de juegos combinatoria: posiciones perdedoras y ganadoras, minimax para juego óptimo</li> <li>• Algoritmos Online vs Offline. Algoritmos probabilísticos.</li> </ul> <p><b>Algoritmos geométricos:</b></p> <ul style="list-style-type: none"> <li>• Representación de puntos, vectores, rectas, segmentos</li> <li>• Detección de puntos alineados, vectores paralelos / perpendiculares, giros horarios / antihorarios</li> <li>• Intersección de dos rectas/segmentos</li> <li>• Área de polígono y verificación de “punto dentro o fuera de polígono”</li> <li>• Compresión de coordenadas</li> <li>• Algoritmos <math>O(N \lg N)</math> para cápsula convexa</li> <li>• Método de Sweep Line</li> </ul> <p><b>Teoría de lenguajes:</b></p> <ul style="list-style-type: none"> <li>• Gramáticas sencillas en BNF</li> </ul>	<ul style="list-style-type: none"> <li>• Problema del viajante de comercio</li> <li>• Problema de la mochila</li> <li>• Problemas de planificación de tareas de manera óptima</li> <li>• Optimizar triangulación de polígonos convexos con algún criterio (ejemplo: mínima suma de perímetros al cuadrado)</li> <li>• Problema de la subsecuencia creciente más larga</li> <li>• Problema de la subsecuencia común más larga</li> <li>• Problema de las ocho reinas y similares en tableros</li> <li>• Problema del par de puntos más cercano entre <math>N</math> en el plano</li> <li>• Dada una posición de Ajedrez o similar juego de tablero, decidir si existe “mate en 2” o similares desafíos</li> </ul>