Olimpiada Boliviana de Informática

Llamado a problemas OBI 2026

Etapa clasificatoria · Comité Científico de la OBI

El Comité Científico de la **Olimpiada Boliviana de Informática (OBI)** invita a exolímpicos, docentes, estudiantes universitarios y profesionales a proponer **problemas de programación** para la **etapa clasificatoria** de los **Niveles 2 y 3** de la OBI 2026, de los que sale la preselección boliviana para la IOI. Las propuestas se envían por correo a obi@jhtan.com, un problema por correo. Consultas a obi@jhtan.com o diegoangulo5@gmail.com.

ETAPA CLASIFICATORIA

Dato	Detalle
Fecha límite de envío	~~Miércoles 7 de octubre de 2026~~ **Domingo 11 de octubre de 2026** (plazo ampliado)
Fecha de la prueba	Del 23 al 25 de octubre de 2026, en línea
Dificultad	Se aceptan problemas de cualquier dificultad. La primera subtarea debe ser accesible para quien recién empieza: para pasar a la departamental basta un punto
Temario	Anexo A de este documento. Sin problemas interactivos

QUÉ DEBE INCLUIR CADA PROPUESTA

Obligatorio:

- **Enunciado bien definido** en español: historia, entrada, salida y restricciones, sin ambigüedades.
- **Puntajes parciales:** al menos **3 subtareas**, o un esquema de puntuación que otorgue puntaje parcial.
- **Solución de 100 puntos:** la idea de la solución y su implementación en C++17, C++20 o Python 3.11.
- **Casos borde:** los casos difíciles o límite que podrían hacer fallar a una solución casi correcta (valores extremos, estructuras degeneradas, desbordes, etc.).

Deseable: el problema **con grader** (quien participa implementa una función que el juez llama), soluciones para cada subtarea y soluciones incorrectas o lentas que deban fallar en una subtarea concreta, para validar los casos de prueba. El formato sugerido para el envío está en el Anexo B.

El Comité Científico se encarga de la redacción final, los casos de prueba, los límites y la carga en la plataforma. Quien propone debe estar dispuesto a **colaborar con el Comité Científico** para ajustar el problema.

CONDICIONES

- Los problemas deben ser **originales**: creados por quien los propone, no publicados antes en ningún medio ni competencia. No se aceptan adaptaciones ni variantes de problemas existentes.
- **Propiedad y uso:** el problema sigue siendo de quien lo propone. Al enviarlo, acepta que solo se use en la OBI y que no se use ni publique en otras competencias.
- Todo el material es **confidencial** hasta el cierre de la etapa clasificatoria, el 25 de octubre.
- Quien proponga no puede ser tutora o tutor académico de participantes de la OBI 2026, y debe declarar si tiene un familiar directo inscrito.
- El Comité Científico elige los problemas y puede adaptarlos. Después de la etapa, la OBI puede publicarlos con crédito a su autora o autor.
- **Reconocimiento:** crédito en el enunciado y en la solución publicada, mención en la web y las redes sociales de la OBI, y un certificado de la OBI por la colaboración. No hay retribución económica.

ANEXO A

Temario de la etapa clasificatoria

Temario de la etapa clasificatoria para los Niveles 2 y 3, tomado del Anexo II de la convocatoria OBI 2026. No se aceptan problemas interactivos.

A.1. Nivel 2

- **Resolución de problemas:** problemas lógicos, aritmética y álgebra.
- **Definición de datos:** tipos de datos, constantes y variables; rangos de `int` y `long long` y desbordamiento.
- **Construcción de algoritmos:** expresiones y asignaciones, entrada y salida, estructuras de control condicionales (if-then-else, switch) e iterativas (for, while, do-while).
- **Números:** series aritméticas, geométricas y especiales; divisibilidad, números primos (prueba de divisores hasta $\sqrt{n}$), descomposición en factores, MCD (algoritmo de Euclides) y mcm, factorial, aritmética modular (suma, resta y multiplicación), división entera y residuo con negativos, conversión de bases.
- **Matemática:** enteros (signo y paridad), fracciones y porcentajes, geometría elemental (punto, vector, polígonos, distancia euclidiana, teorema de Pitágoras) y sistemas de numeración.
- **Estructuras discretas:** conjuntos, tablas de verdad, conteo (regla de suma y producto, progresiones, Fibonacci), coeficientes binomiales y triángulo de Pascal.
- **Fundamentos de programación:** funciones y paso de parámetros, descomposición estructurada, estrategias de resolución y depuración, arreglos (incluso multidimensionales), números reales y problemas de precisión del punto flotante.
- **Cadenas:** recorrido carácter por carácter, código ASCII, palíndromos, subcadenas, búsqueda ingenua de patrones, conversión entre cadena y número.
- **Técnicas sobre arreglos:** sumas de prefijos, arreglo de diferencias, dos punteros, ventana deslizante, conteo de frecuencias (counting sort).
- **Simulación y búsqueda completa:** simulación paso a paso, recorrido de matrices y cuadrículas (vecinos en 4 y 8 direcciones), generación de subconjuntos y permutaciones.
- **Algoritmos:** notación O y clases de complejidad, fuerza bruta, algoritmos golosos, divide y vencerás, búsqueda secuencial y binaria (`lower_bound`, `bisect`), algoritmos de ordenamiento y ordenamiento con criterio propio (pares, tuplas).
- **Estructuras de datos:** arreglos, pilas, colas, conjuntos y mapas.

A.2. Nivel 3

Incluye todo el Nivel 2, también los siguientes temas:

- **Recursividad:** definiciones recursivas, recursividad y memoización.
- **Números:** criba de Eratóstenes, factorización con criba y exponenciación rápida; uso de fracciones para cálculos precisos.
- **Grafos:** representación (lista y matriz de adyacencia), BFS y DFS, BFS en cuadrículas (laberintos, distancia mínima), relleno por inundación (flood fill) y componentes conexas; árboles y sus propiedades básicas.
- **Técnicas:** búsqueda binaria sobre la respuesta; cola de prioridad (heap) y su uso en algoritmos golosos.
- **Programación dinámica:** escaleras y caminos en cuadrícula, mochila 0/1, cambio de monedas, subsecuencia creciente más larga (LIS) en $O(n^2)$ y subsecuencia común más larga (LCS).
- **Operaciones de bits:** AND, OR, XOR, desplazamientos y enumeración de subconjuntos con máscaras.

Además:

- **Estructuras discretas:** funciones (inyectiva, sobreyectiva, biyectiva), relaciones y orden lexicográfico, técnicas de prueba (directa, contraposición, contradicción, inducción), inclusión-exclusión, principio del palomar, teoría de grafos (caminos, ciclos, conectividad, grafos eulerianos y hamiltonianos, bipartitos y planares, solo como definiciones) y árboles de expansión.
- **Fundamentos de programación:** tipos de datos abstractos (struct, objetos), memoria estática y en pila, punteros y referencias, backtracking recursivo.
- **Técnicas:** pila y cola monótonas, meet in the middle, compresión de coordenadas, análisis amortizado.
- **Programación dinámica:** sobre DAG, intervalos, máscaras de bits, árboles y dígitos; LIS en $O(n \log n)$; reconstrucción de la solución.
- **Grafos:** orden topológico y caminos de Euler, componentes conexas, puentes y puntos de articulación, componentes fuertemente conexas, caminos más cortos (BFS 0-1, Dijkstra con cola de prioridad, Bellman-Ford, Floyd-Warshall), árbol de expansión mínima (Prim, Kruskal), emparejamiento bipartito máximo; diámetro y centro de un árbol, recorridos preorden, inorden y postorden, ancestro común más bajo (LCA) con binary lifting.
- **Estructuras de datos:** conjuntos y mapas ordenados (`std::set`, `std::map`), Union-Find con compresión de caminos y unión por rango, sparse table (RMQ), Fenwick tree, Segment tree con propagación perezosa (lazy), descomposición en raíz cuadrada, árboles binarios balanceados y tries.
- **Cadenas:** algoritmo KMP y hashing de cadenas.
- **Combinatoria y juegos:** combinatoria; juegos con posiciones ganadoras y perdedoras, y minimax.
- **Aritmética:** enteros grandes (suma, resta y multiplicación simple) y exponenciación rápida modular.
- **Geometría con coordenadas enteras:** producto cruz y producto punto, orientación (giro horario o antihorario), colinealidad, intersección de segmentos, área de polígonos (fórmula del zapatero) y

punto dentro de un polígono.

A.3. Temas excluidos

- Álgebra lineal (multiplicación y exponenciación de matrices, eliminación gaussiana) y transformada rápida de Fourier (FFT).
- Cálculo, estadística y resolución de recurrencias.
- Geometría con dualidad punto-línea, diagramas de Voronoi o triangulación de Delaunay, intersecciones con círculos, trigonometría, geometría en 3D o cálculos que exijan manejar errores de precisión.
- Juegos de Nim con XOR y números de Sprague-Grundy.
- Estructuras avanzadas como link-cut trees o estructuras bidimensionales anidadas.

ANEXO B

Formato sugerido para el envío

B.1. Correo

- **Para:** obi@jhtan.com, un problema por correo.
- **Asunto:** `[CFT OBI 2026] Nombre del problema - Nombre del autor`.
- **Adjunto:** un archivo `.zip` con la estructura de abajo, llamado con el nombre corto del problema (por ejemplo, `caminos-andinos.zip`).

B.2. Estructura del archivo (formato de tareas de la OBI)

Archivo o carpeta	Contenido
`autor.txt`	Nombre, correo, institución o afiliación, y una declaración de que el problema es original y de que no hay conflicto de interés (o cuál es)
`statement/es.md`	Enunciado en español (Markdown o PDF) con las secciones de la tabla siguiente
`grader/`	*Muy recomendado:* el grader (`grader.cpp` y, si es posible, `grader.py`) y la plantilla con la firma de la función que implementa quien participa
`solution/correcta.cpp`	Solución de 100 puntos, en C++17, C++20 o Python 3.11 (`.py`)
`solution/idea.md`	Idea de la solución y su complejidad
`solution/subtarea-N.cpp`	*Deseable:* una solución para cada subtarea
`solution/incorrecta-*.cpp`	*Deseable:* soluciones incorrectas o lentas, con la subtarea en la que deben fallar
`tests/`	Casos borde: archivos `.in` con su `.out`, o un `casos-borde.md` que los describa

B.3. Secciones del enunciado

Sección	Qué incluye
Historia	El contexto del problema, breve y sin ambigüedades
Entrada	Parámetros de la función del grader, o el formato exacto de la entrada si no tiene grader
Salida	Lo que devuelve la función, o el formato exacto de la salida si no tiene grader
Restricciones	Límites de todas las variables
Subtareas	Al menos 3, cada una con su puntaje (suman 100) y sus restricciones adicionales
Ejemplos	Al menos un ejemplo de entrada y salida, con su explicación

Todos los problemas de la OBI usan grader: es **mucho mejor** enviarlo con grader.