



Preparación para el concurso

Páginas para entrenar, que tienen evaluador en línea:

<http://teddy.itc.mx/> (tiene problemas sencillos en español). <https://omegaup.com/> (tiene problemas complicados en español) utilizados por concursos de programación a nivel nacional como el CONACUP y la Olimpiada Mexicana de Informática (OMI). <http://uva.onlinejudge.org/> (tiene todo tipo de problemas en inglés) utilizado por el concurso internacional organizado por IBM y ACM International Collegiate Programming Contest (ICPC). <https://code.google.com/codejam/> (tiene todo tipo de problemas en inglés) utilizado por el concurso internacional de Google CodeJam.

Algunas otras son:

<http://www.codechef.com/>

<http://www.spoj.com/>

<http://www.topcoder.com/>

El evaluador en línea, prueba tu código con ciertos casos de prueba y te informa si es correcto o incorrecto, algunos evaluadores dicen si el problema es de Tiempo, Compilación o simplemente el Resultado es diferente del esperado.

Problemas IEEEExtreme (concursos anteriores): 1.0, 2.0 y 3.0 en la página: <http://olympiad.bytefreaks.net/events/ieeextreme-problem-sets>

4.0 en la página: <http://hazardhq.com/ieeextreme4/> .5.0 y 6.0: <https://docs.google.com/file/d/0B43jYE3dXFKcakVCam15bDdZbEE/edit?usp=sharing>

NOTAS:

Es importante entender la lectura de los datos de entrada y la salida de los resultados. En <http://teddy.itc.mx/> se realiza leyendo un archivo `data.in` y escribiendo un archivo `data.out`. En <https://omegaup.com/>, <http://uva.onlinejudge.org/> e IEEEExtreme se utiliza la entrada por teclado y salida en pantalla estándar. En <https://code.google.com/codejam/> se descarga un archivo `*.in` y solo se necesita adjuntar el archivo de salida `*.out`.

Respetar el formato de salida (puntos, comas, espacios, mayúsculas, minúsculas, etc.)

Lee cuidadosamente los límites de cada variable en el problema para luego definirlos con el tipo correcto en el programa. Ejemplo n es un entero positivo donde $0 < n < 2^{32}$ no se puede utilizar en JAVA un "int" entonces se debe usar un BigInteger.

Prueba tu programa con los ejemplos de prueba, si pasa todas las pruebas crea nuevas pruebas para verificar. Los ejemplos de prueba vienen dentro del problema, pero tu programa se probará con más casos de prueba (en ocasiones los mismos

ejemplos de prueba + algunos casos extra). Los organizadores de un concurso buscan las situaciones que harán de un programa fallar ya sea en tiempo de ejecución o por exceso de memoria usada, tu tarea como competidor es visualizar estos peores casos y evitarlos con algoritmos bien diseñados.

Si algún problema no pasa todos los casos de prueba y no logras encontrar el error, pasa al siguiente problema. No desperdices demasiado tiempo en un problema.

Tipos Frecuentes de Problemas:

- Problemas de Búsqueda
- Problemas de Grafos
- Problemas de Geometría
- Problemas de programación dinámica
- Problemas Triviales.

Clasifica los problemas e intenta resolver primero el mas fácil.