

Introducción al Campamento

Pablo Zimmermann

Universidad Nacional de Rosario

11th Caribbean Camp

- 1 Presentación
- 2 Cuestiones Prácticas
- 3 Definiciones Básicas
 - ACM ICPC
 - Algoritmos
 - Complejidad Algorítmica

- Hay un Dropbox público donde publicaré todo lo dado (<http://bit.ly/tccuba>)
- Habrá 5 clases teóricas
- Habrá 7 competencias (1 individual que ya hicieron y 6 grupales), cada una con su solucionario
- Los problemas serán principalmente de Codeforces, SPOJ y UVA (intentaré no usar nada después del 2014-2015)
- Como juez, usaremos vjudge o codeforces (discusión)
- Los lenguajes permitidos para enviar son C, C++, C++11, Java y Python
- Aparecerán eventualmente problemas interactivos y constructivos
- Usaremos muchos códigos de “El Diego”

Contenidos

- 1 Presentación
- 2 Cuestiones Prácticas
- 3 **Definiciones Básicas**
 - **ACM ICPC**
 - Algoritmos
 - Complejidad Algorítmica

ACM ICPC

- Es una competencia de programación en equipos
- 3 miembros de la misma universidad
- 5 horas
- 10-13 Problemas
- Hacer un algoritmo por cada problema

Contenidos

- 1 Presentación
- 2 Cuestiones Prácticas
- 3 Definiciones Básicas**
 - ACM ICPC
 - Algoritmos**
 - Complejidad Algorítmica

Algoritmos

“Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.”

Algoritmo Computacional

Es un algoritmo que puede ser realizado por una computadora.

Ejemplo: Calcular si un número dado es primo.

En este caso podríamos proceder por ejemplo verificando si algun natural menor a él distinto de 1 lo divide.

Contenidos

- 1 Presentación
- 2 Cuestiones Prácticas
- 3 Definiciones Básicas**
 - ACM ICPC
 - Algoritmos
 - Complejidad Algorítmica**

Para decidir cuan bueno o eficiente es un algoritmo vamos a medir su complejidad. La complejidad de un algoritmo nos dice cuanto usa los recursos de nuestra computadora. Los recursos más importantes con los que cuenta una computadora a la hora de ejecutar un programa son dos:

- Memoria: La máxima cantidad de memoria que usa el programa al mismo tiempo. Si usa varias veces la misma memoria se cuenta una sólo vez.
- Tiempo: Cuánto tarda en correr el algoritmo. Se mide en cantidad de operaciones básicas (sumas, restas, asignaciones, etc).

Memoria

El uso de la memoria lo medimos por la mayor cantidad de memoria que usa un programa al mismo tiempo.

Si un programa:

- Utiliza 200MB de memoria
- Libera 50MB
- Ocupa otros 150MB

La memoria que utiliza el programa es 300MB y no 350MB ya que nunca utiliza 350MB al mismo tiempo.

- En las World Finals tenemos 2GB de memoria
- En la final regional tenemos 1GB de memoria (pruebenlo)
- Si nos excedemos, nos avisará con un “Memory Limit Exceeded”

Tiempo

La complejidad temporal es el tiempo total que tarda en ejecutarse en su totalidad.

Como medimos el uso del tiempo?

- El uso del tiempo es difícil de medir. Lo aproximamos como la cantidad de operaciones básicas que ejecuta el programa, como pueden ser sumas, restas, asignaciones, etc.
- Este número es complicado de calcular y por lo general nos interesa más tener una idea de cómo crece esta complejidad a medida que crece la entrada del problema, por eso hablamos de órdenes de complejidad.

O grande

El orden de complejidad que más se suele usar en la ACM-ICPC es el de la O grande.

Este orden acota superiormente un algoritmo en su peor caso de ejecución.

Ejemplo: Con el algoritmo para ver si un número era primo de arriba, a lo sumo hace n operaciones, por lo que lo acotamos por $O(n)$.

Como las constantes son difíciles de calcular, las vamos a ignorar, por lo que $O(2n) = O(n)$.

La práctica me dice que en general una máquina hace a lo sumo 10^9 operaciones por segundo

- En general nos dan 1 o 2 segundos.
- 10^7 operaciones está perfecto
- 10^8 probablemente pase el límite de tiempo
- 10^9 por otro lado probablemente no funcione
- 10^{10} seguro que no va a pasar el límite
- Cuando tarda mucho tiempo nos avisa con un “Time Limit Exceeded”

Si tenemos un problema, donde $N \leq 10^5$, que complejidades podemos intentar??

La práctica me dice que en general una máquina hace a lo sumo 10^9 operaciones por segundo

- En general nos dan 1 o 2 segundos.
- 10^7 operaciones está perfecto
- 10^8 probablemente pase el límite de tiempo
- 10^9 por otro lado probablemente no funcione
- 10^{10} seguro que no va a pasar el límite
- Cuando tarda mucho tiempo nos avisa con un “Time Limit Exceeded”

Si tenemos un problema, donde $N \leq 10^5$, que complejidades podemos intentar??

- $O(N)$, $O(N \lg(N))$, $O(N\sqrt{N})$
- $\cancel{O(N^2)}$