

# TP1: Indiana Jones en búsqueda de la complejidad esperada

**Fecha de entrega:** viernes 2 de septiembre, hasta las 18:00 hs.

Este trabajo práctico consta de varios problemas y para aprobar el mismo se requiere aprobar todos los problemas. La nota final será un promedio ponderado de las notas finales de los ejercicios y el trabajo práctico se aprobará con una nota de 5 (*cinco*) o superior. De ser necesario (o si el grupo lo desea), el trabajo podrá reentregarse una vez corregido por los docentes y en ese caso la reentrega deberá estar acompañada por un *informe de modificaciones*. Este informe deberá detallar brevemente las diferencias entre las dos entregas, especificando los cambios, agregados y/o partes eliminadas. Cualquier cambio que no se encuentre en dicho informe podrá no ser tenido en cuenta en la corrección de la reentrega. Lo grupos deben ser de tres o cuatro personas.

Para cada ejercicio se pide encontrar una solución algorítmica al problema propuesto y desarrollar los siguientes puntos:

1. Describir detalladamente el problema a resolver dando ejemplos del mismo y sus soluciones.
2. Explicar de forma clara, sencilla, estructurada y concisa, las ideas desarrolladas para la resolución del problema. Para esto se pide utilizar pseudocódigo y lenguaje coloquial combinando adecuadamente ambas herramientas (**¡sin usar código fuente!**). Se debe también justificar por qué el procedimiento desarrollado resuelve efectivamente el problema.
3. Deducir una cota de complejidad temporal del algoritmo propuesto (en función de los parámetros que se consideren correctos) y justificar por qué el algoritmo desarrollado para la resolución del problema cumple la cota dada. Utilizar el modelo uniforme salvo que se explicita lo contrario.
4. Dar un código fuente claro que implemente la solución propuesta. El mismo no sólo debe ser correcto sino que además debe seguir las *buenas prácticas de la programación* (comentarios pertinentes, nombres de variables apropiados, estilo de indentación coherente, modularización adecuada, etc.). Se deben incluir las partes relevantes del código como apéndice del informe impreso entregado.
5. Realizar una experimentación computacional para medir la performance del programa implementado. Para ello se debe preparar un conjunto de casos de test que permitan observar los tiempos de ejecución en función de los parámetros de entrada. Deberán desarrollarse tanto experimentos con instancias aleatorias (detallando cómo fueron generadas) como experimentos con instancias particulares (de peor/mejor caso en tiempo de ejecución, por ejemplo). Se debe presentar **adecuadamente** en forma gráfica una comparación entre los tiempos medidos y la complejidad teórica calculada y extraer conclusiones de la experimentación.

Respecto de las implementaciones, se acepta cualquier lenguaje que permita el cálculo de complejidades según la forma vista en la materia. Además, debe poder compilarse y ejecutarse correctamente en las máquinas de los laboratorios del Departamento de Computación. La cátedra recomienda el uso de C++ o Java, y se sugiere consultar con los docentes la elección de otros lenguajes para la implementación. **Solo se permite utilizar 1 lenguaje para todo el TP.**

La entrada y salida de los programas **deberá hacerse por medio de la entrada y salida estándar del sistema**. No se deben considerar los tiempos de lectura/escritura al medir los tiempos de ejecución de los programas implementados.

Deberá entregarse un informe impreso que desarrolle los puntos mencionados. Por otro lado, deberá entregarse el mismo informe en formato digital acompañado de los códigos fuentes desarrollados e instrucciones de compilación, de ser necesarias. Estos archivos deberán enviarse a la dirección `algo3.dc@gmail.com` con el asunto "*TP 1: Apellido.1, ..., Apellido.n*", donde  $n$  es la cantidad de integrantes del grupo y *Apellido.i* es el apellido del  $i$ -ésimo integrante.

## Problema 1: Cruzando el puente

Indiana Jones disfruta tranquilamente de su retiro. Pasa los días estudiando algoritmos nuevos para descifrar los secretos de civilizaciones antiguas. En un momento descubre que un mapa podría liderarlo hacia una antigua civilización persa que podría tener resuelta la pregunta más importante de la computación, es  $P=NP$ ?

Indy viaja hacia el medio oriente en donde comienza la expedición hacia los tesoros de la civilización junto con su grupo de arqueólogos. Una tribu local los ayuda para acompañarlos a través del bosque en donde se encuentra la fortaleza persa.

En el medio de la noche se encuentra con un puente colgante (de estructura bastante dudosa) para cruzar un río. El grupo tiene una sola linterna que se debe utilizar para cruzar el puente. Debido al estado del puente no se arriesgarían que crucen más de dos personas a la vez.

El problema es que la tribu que lo acompaña es conocida por su canibalismo! Por lo tanto tienen una restricción de que en ningún momento puede haber en un lado del puente más caníbales que arqueólogos.

**Si no hay arqueólogos pueden haber caníbales.**

Cada persona tiene una velocidad a la cual se mueve, y si dos personas cruzan el puente lo cruzarán a la velocidad del más lento (el que tarde más en cruzarlo).

Indy quiere que todos crucen el puente lo más rápido posible. Podremos ayudarlo?

**Formato de entrada:** La primera línea consta de dos enteros positivos  $N$  y  $M$ , donde  $N$  es la cantidad de arqueólogos y  $M$  es la cantidad de caníbales. La segunda línea contine  $N$  enteros indicando el **tiempo** que tardan en cruzar el puente los arqueólogos. La tercer línea contine  $M$  enteros indicando el **tiempo** que tardan en cruzar el puente los caníbales. La entrada contará con el siguiente formato:

```
N M
V_1 V_2 ... V_N
W_1 W_2 ... W_M
```

**Formato de salida:** La primera línea de la salida debe contener un número  $T$  indicando el mínimo tiempo necesario para que todos crucen el puente. En caso de no ser posible imprimir  $-1$ . La salida tendrá el siguiente formato:

```
T
```

**Restricciones de entrada:**

```
1 ≤ N + M ≤ 6
1 ≤ V_i ≤ 10^6
1 ≤ W_i ≤ 10^6
```

## Problema 2: Problemas en el camino

El equipo logro cruzar el puente satisfactoriamente y ya se encuentra dentro de una fortaleza antigua.

Al entrar, nuestro grupo de aventureros caminan por los pasillos hasta llegar a una habitación amplia con una puerta y algunos esqueletos tirados en el piso.

Al tratar de abrir la puerta se dan cuenta que la puerta esta cerrada con llave.

En el medio de la habitación hay una balanza de dos platillos, en el plato de la izquierda se encuentra la llave para abrir la siguiente puerta, el otro esta vacío.

Al tomar la llave, la balanza cambia el equilibrio, se cierra la entrada a la habitación, las paredes se empiezan a mover y la llave no sirve para destrabar la puerta.

Rapidamente observan que debajo de la balanza se encuentra un conjunto de pesas. Curiosamente todas tienen un escritas una potencia de 3 distinta que indican el peso de cada una.

Entre los expedicionarios uno tiene una balanza digital para pesar la llave que retiraron. Quieren recomponer el equilibrio de la balanza utilizando las pesas encontradas.

Ayudemoslo a reestablecer la balanza antes de que las paredes terminen aplastando a los integrantes de la expedición!

**Formato de entrada:** La entrada consiste en un entero  $P$ , el peso de la llave. La entrada contará con el siguiente formato:

$P$

**Formato de salida:** La primer línea consiste en dos enteros  $S$  y  $T$ . La segunda línea consiste de  $S$  números, las pesas que se colocan en el plato izquierdo ordenadas de menor a mayor. La tercer línea consiste de  $T$  números, las pesas que se colocan en el plato derecho ordenadas de menor a mayor.

$S \ T$   
 $I_1 \ I_2 \ \dots \ I_S$   
 $D_1 \ D_2 \ \dots \ D_T$

**Restricciones de entrada:**

$$1 \leq P \leq 10^{15}$$

La complejidad temporal deberá ser a lo sumo  $\mathcal{O}(\sqrt{P})$

### Problema 3: Guardando el tesoro

Una vez equilibrada la balanza, las paredes vuelven a su lugar. Con la llave obtenida logran abrir la puerta que estaba trabada.

Al iluminar la habitación se encuentran con una sala llena de tesoros, monedas, calices, joyas, de todo. Cada tipo de tesoro tiene un valor y un peso. Si bien el objetivo principal de la expedición era otro, nunca viene mal armarse de algún *recuerdo*.

Tienen varias mochilas disponibles para llevarse los tesoros y cada una tiene un máximo de peso que puede llevar.

Cómo podrían guardar las cosas para llevarse el mayor valor posible?

**Formato de entrada:** La primera línea consta de dos enteros positivos  $M$  y  $N$ , que indica la cantidad de mochilas y la cantidad de tipos de tesoros. A esta línea le sigue una línea con  $M$  números, uno para cada mochila, cada uno corresponde a un entero  $K_i$  indicando la capacidad. Las siguientes  $N$  líneas contienen tres enteros  $C_i$ ,  $P_i$  y  $V_i$  indicando la cantidad de tesoros que hay de ese tipo, cuánto pesa cada uno, y cuál es el valor. La entrada contará con el siguiente formato:

$M \ N$   
 $K_1 \ K_2 \ \dots$   
 $C_1 \ P_1 \ V_1$   
 $C_2 \ P_2 \ V_2$   
 $\dots$   
 $C_N \ P_N \ V_N$

**Formato de salida:** La primera línea de la salida debe contener un número  $S$  indicando el valor total de los tesoros guardados. Las siguientes  $M$  líneas deben contener primero un número  $Q$  indicando la cantidad de tesoros, seguido de  $Q$  números indicando el tipo de tesoro (Puede haber repetidos si hay suficiente cantidad de algún tipo de tesoros). Los tipos de tesoros están numerados del 1 al  $N$ , según su orden en el archivo de entrada. La salida tendrá el siguiente formato:

$S$   
 $Q_1 \ T_{11} \ T_{12} \ \dots \ T_{1Q_1}$   
 $Q_2 \ T_{21} \ T_{22} \ \dots \ T_{2Q_2}$   
 $\dots$   
 $Q_M \ T_{M1} \ T_{M2} \ \dots \ T_{MQ_M}$

**Restricciones de entrada:**

$$1 \leq M \leq 3$$
$$1 \leq K_i \leq 50$$
$$1 \leq \sum C_i \leq 50$$
$$1 \leq P_i \leq 100$$
$$1 \leq V_i \leq 10^6$$

La complejidad temporal deberá ser a lo sumo  $\mathcal{O}((\sum K_i)^M \times (\sum C_i))$