UNIVERSIDAD ORT
Uruguay

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

EVALUACIÓN	Obligatorio	**GRUPO**	Todos	**FECHA**	29/8
MATERIA	Estructura de Datos y Algoritmos 2				
CARRERA	Ingeniería en Sistemas				
CONDICIONES	- Puntos: Máximo: 30 Mínimo: 1 LA ENTREGA SE REALIZA EN FORMA ONLINE EN ARCHIVO NO MAYOR A 40 MB EN FORMATO ZIP, RAR O PDF. **IMPORTANTE:** - Inscribirse - Formar grupos de hasta dos personas. - Subir el trabajo a Gestión antes de la hora indicada, ver hoja al final del documento: “RECORDATORIO”				

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Obligatorio

El obligatorio de Estructuras de Datos y Algoritmos 2, para el semestre 2 de 2023, está compuesto por un conjunto de **10** problemas a resolver.

El desarrollo del obligatorio:

- Se deberá realizar en grupos de **2 estudiantes**.
- Se podrá utilizar tanto C++ (C++11) como Java (jdk8).
  Para asegurar que utilizan C++11 deberán tener instalado el compilador C++ de GNU e invocar la siguiente orden: **`g++ -std=c++11 ...`**
- No se puede usar ninguna librería, clase, estructura, función o variable no definida por el estudiante, a excepción de <iostream>, <string> y <cstring> para C++. y System.in, System.out y String para Java.
- Si se quiere usar algo que no sea lo listado anteriormente, consultar previamente al profesor.
- Se deberá utilizar el formato de ejecución trabajado durante el curso:
  `miSolucion < input > output`
- La cátedra proveerá un conjunto de casos de prueba, pares de entrada y salida esperada. Se deberá comparar la salida de la solución contra una salida esperada.
- **De no cumplir las condiciones anteriores el ejercicio se considera invalido, por lo tanto 0 puntos.**

La entrega:

- El formato de entrega se encuentra definido en este enlace.

La corrección:

- La corrección implica la verificación contra la salida esperada, así también como la corrección del código fuente para verificar que se cumplan los requerimientos solicitados (órdenes de tiempo de ejecución, usos de estructuras o algoritmos en particular, etc.).
- Se verificarán TODOS los casos de prueba. Si el programa no termina para un caso de prueba, puede implicar la pérdida de puntos.
- Si el código fuente se encuentra en un estado que dificulta la comprensión, podrá perder puntos.
- Los códigos dados en clase como algoritmos encontrados de otras fuentes serán usados como referencia. **Considerándose copia el uso de ellos.**
- **Se utilizará MOSS para detectar copias.**

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

La defensa:

- Luego de la entrega, se realizará una defensa de autoría a cada estudiante **de manera individual**.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

1. Restaurante

Nombre de archivo: ejercicio1.cpp/Ejercicio1.java

Descripción del problema

Tienes un restaurante que ofrece varios tipos de platos a los clientes. Quieres saber cuáles son los platos más populares entre los clientes. Para eso, decidiste utilizar una tabla hash para llevar la cuenta de cada plato ordenado.

Entrada

La primera línea contiene un entero N $(1 \leq N \leq 10^5)$, que representa el número de órdenes recibidas.
Las siguientes N líneas contienen una cadena S_i $(1 \leq |S_i| \leq 100)$, que representa el nombre del plato ordenado.

Salida

Debes imprimir todos los platos ordenados por la cantidad de veces que han sido ordenados (de más a menos veces ordenado). En caso de empate, ordenar alfabéticamente.

Restricciones

- Orden temporal: $O(N + K \log K)$ siendo K la cantidad de platos diferentes

Ejemplo de Entrada

7
Pizza
Hamburguesa
Tacos
Pizza
Hamburguesa
Ensalada
Tacos

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Ejemplo de Salida

Hamburguesa
Pizza
Tacos
Ensalada

2. Las Pelis

Nombre de archivo: ejercicio2.cpp/Ejercicio2.java

Descripción del problema

Supón que estás trabajando en un sistema de recomendaciones de películas. Cada usuario puede calificar una película con una puntuación entre 1 y 5. Además, cada película pertenece a un género. Tu objetivo es encontrar, para cada género, la película con la calificación promedio más alta.

Entrada

- La primera línea contiene un entero G, $1 <= G <= 10$, que representa el número de géneros de películas del sistema.
- Las siguientes G líneas contienen los nombres de los géneros.
- La siguiente línea contiene un entero P, $1 <= P <= 10^5$, que representa la cantidad de películas en el sistema.
- Las siguientes P líneas contienen el ID de la película y a que género pertenecen.
- La siguiente línea contiene un entero N, $1 \leq N \leq 10^5$, que representa el número de calificaciones que se han dado.
- Las siguientes N líneas contienen dos campos separados por espacios: el ID de la película M, $1 \leq M \leq 10^5$ y la puntuación P, $1 \leq P \leq 5$.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Salida

Para cada género (de la misma manera que fueron dados), imprime el ID de la película con la calificación promedio más alta. En caso de empate, imprimir el ID de película más pequeño.

Restricciones

- Orden Temporal: $O(N)$ caso promedio

3. Urgencias

Nombre de archivo: ejercicio3.cpp/Ejercicio3.java

Descripción del problema

Imagínate que estás desarrollando un sistema para una clínica que recibe un flujo constante de pacientes. Cada paciente llega en un momento diferente y con un nivel diferente de urgencia. Tu objetivo es implementar un sistema que priorice a los pacientes en función de su nivel de urgencia y el orden de llegada.

Entrada

- La primera línea contiene un entero N, que representa el número de pacientes.
- Las siguientes N líneas contienen tres enteros P, T y U, que representan el paciente, el tiempo de llegada y el nivel de urgencia del paciente, respectivamente.

Salida

Imprime el orden en que los pacientes serán atendidos, comenzando con el paciente que tiene el nivel de urgencia más alto (mayor número de urgencia). En caso de empate en urgencia, da prioridad al paciente que llegó primero. En caso de empate de urgencia y tiempo de llegada, se toma al paciente que se ingresó primero.

Restricciones

- Temporal: $O(N \log N)$ peor caso
- Espacial: N

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

- No usar AVL

Ejemplo de Entrada

```
4
10 0 3
15 5 2
7 2 3
9 8 1
```

Ejemplo de Salida

```
10
7
15
9
```

4. Preparación de Platos

Nombre de archivo: ejercicio4.cpp/Ejercicio4.java

Descripción del problema

Eres el chef principal de un restaurante de alta cocina. En tu menú, hay ciertos platos que requieren que otros platos se preparan antes. Por ejemplo, antes de servir el "Pollo Rostizado", necesitas tener preparado el "Aderezo de Hierbas". Esta dependencia entre platos puede verse como un gráfico dirigido, donde un plato que depende de otro está conectado por una arista dirigida.

Tu tarea es determinar un orden en el que preparar los platos de manera que todas las dependencias sean respetadas. Si no es posible preparar los platos debido a que hay ciclos en las dependencias, deberán informarlo.

Si hay múltiples soluciones posibles debido a platos con igual prioridad, devuelva la que primero ordene los platos en orden alfabético.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Entrada

- La primera línea contiene dos enteros P y D, que representa el número total de platos y el número de dependencias entre platos respectivamente.
- Las siguientes P líneas contienen una cadena S, el nombre del plato.
- Las siguientes D líneas contienen dos cadenas A y B, que representan que el plato A debe ser preparado antes que el plato B.

Salida

Si es posible preparar los platos respetando las dependencias, imprime en orden el nombre de los platos. Esta salida debe ser única, respetando las dependencias y ordenando alfabéticamente los platos con igual prioridad.
Si no es posible preparar los platos debido a ciclos en las dependencias, imprime "CICLO DETECTADO".

Restricciones

- Temporal: $O(P \log(P) + D)$ peor caso

Ejemplo de Entrada

4 3
AderezoDeHierbas
PolloRostizado
PanDeAjo
SopaDeVerduras
AderezoDeHierbas PolloRostizado
PanDeAjo SopaDeVerduras
SopaDeVerduras PolloRostizado

Ejemplo de Salida

AderezoDeHierbas
PanDeAjo
SopaDeVerduras
PolloRostizado

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

5. Interconexiones Planetarias

Nombre de archivo: ejercicio5.cpp/Ejercicio5.java

Descripción del problema

Eres el capitán de una nave espacial y te encuentras en una galaxia con múltiples sistemas planetarios interconectados por portales espaciales. Cada portal conecta dos planetas y tiene un costo de energía asociado para viajar a través de él.

Tu objetivo es determinar la cantidad mínima de energía requerida para viajar desde un planeta de origen a un planeta de destino, utilizando los portales.

Entrada

- La primera línea contiene dos enteros N y M, que representan el número de planetas y el número de portales, respectivamente.
- Las siguientes M líneas contienen tres enteros cada una: A, B y C que representan un portal entre el planeta A y el planeta B con un costo de C.
- La última línea contiene dos enteros, O y D, que representan el planeta de origen y el planeta de destino, respectivamente.

Salida

Imprime la cantidad mínima de energía requerida para viajar desde el planeta de origen al planeta de destino. Si no es posible llegar al planeta de destino desde el planeta de origen, imprime -1.

Restricciones

- Temporal: O((M+N) log(N))

Ejemplo de Entrada

```
5 6
1 2 10
2 3 5
3 4 20
4 5 10
1 3 30
```

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

3 5 15
1 5

Ejemplo de Salida

30

6. Carreteras

Nombre de archivo: ejercicio6.cpp/Ejercicio6.java

Descripción del problema

Supongamos que eres un ingeniero de redes y estás a cargo de establecer conexiones de comunicación óptimas entre N ubicaciones. Cada ubicación está representada por un nodo y las conexiones se pueden establecer entre estos nodos. Sin embargo, establecer una conexión entre dos nodos tiene un costo asociado.

Tu tarea es determinar el costo mínimo para establecer una red de comunicación que conecte todas las ubicaciones.

Entrada

- La primera línea contiene dos enteros: N (número de ubicaciones) y M (número de posibles conexiones entre ubicaciones).
- Las siguientes M líneas contienen tres enteros cada una: A, B y C, que indican que se puede establecer una conexión directa entre las ubicaciones A y B con un costo de C.

Salida

Imprime un solo entero que representa el costo mínimo para establecer la red de comunicación que conecta todas las ubicaciones.

Restricciones

- Temporal: $O(|M| \log |N|)$

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Ejemplo de Entrada

4 5
1 2 3
1 3 2
1 4 5
2 3 1
3 4 4

Ejemplo de Salida

7

7. Mínima suma de sub-enteros

Nombre de archivo: ejercicio7.cpp/Ejercicio7.java

Descripción del problema

Dado un número 'N', se busca encontrar dos números 'N1' y 'N2' que se forman con los dígitos de N, de forma tal de que todos los dígitos de N1 y N2 forman N, y la suma de N1 y N2 es mínima.

Entrada

Un entero 'N' que representa el número de entrada.

Salida

Un entero que representa la suma de los dos números('N1' y 'N2') que se forman con los dígitos de 'N', de forma tal de que todos los dígitos de N1 y N2 forman N, y la suma de N1 y N2 es mínima.

Restricciones

- Orden Temporal: O(N Log N) con N la cantidad de dígitos del número de entrada.
- Orden Espacial: O(N) con N la cantidad de dígitos del número de entrada.
- Otras: 1 <= N <= 10^9.
- Realizar con **Greedy**

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Ejemplo de Entrada

1234

Ejemplo de Salida

37

> Dado que yo puedo separar el 1234 (N) en dos números 24 (N1) y 13 (N2), estos suman 37 (respuesta final)

Otros ejemplos de entrada y salida:

Entrada	Salida
687	75
4325	59
1891	37

8. Montañas y Picos

Nombre de archivo: ejercicio8.cpp/Ejercicio8.java

Descripción del problema

Estás trabajando con un geólogo particularmente interesado en estudiar los picos de la Cordillera de los Andes. La cadena montañosa se representa con la altura de cada una de las montañas que la componen. Tu tarea es identificar los picos en la cadena de montañas. Un pico se define como un punto más altos que sus vecinos inmediatos; además, un pico se puede encontrar en cualquiera de sus extremos

Entrada

La entrada consta de dos partes:

- Un entero N (1 <= N <= 10^5) que representa la cantidad de montañas de la cadena
- N enteros que representan la altura de cada montaña, h (h > 0)

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Salida

La salida debe ser P líneas representando la altitud de cada uno de los picos en la cadena montañosa considerando el orden en que aparecen originalmente en la cadena de montañas.

Restricciones

Siendo P la cantidad de picos encontrada y N la cantidad de montañas:

- Orden Temporal: O(N * Log N)
- Orden Espacial: O(N)
- Otras: se debe implementar utilizando la estrategia **Divide & Conquer**

Ejemplo de Entrada

```
3
356
200
200
```

Ejemplo de Salida

```
356
```

9. La Play

Nombre de archivo: ejercicio9.cpp/Ejercicio9.java

Descripción del problema

Después del gran éxito del PolyStation 5, Chony ha lanzado su versión más compacta y elegante: el PS5 Slim. A pesar de su diseño más delgado, viene con la misma potencia que su predecesora. Para celebrar el lanzamiento, Chony ha decidido ofrecer a sus usuarios leales una selección de juegos de su extenso catálogo. Sin embargo, dado que el PS5 Slim tiene un disco duro con capacidad limitada, los usuarios deben elegir sabiamente qué juegos instalar.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Como jugador, estás emocionado por probar algunos de los mejores títulos disponibles, pero también eres consciente del espacio y quieres asegurarte de obtener el máximo valor en términos de popularidad y recency de los juegos que elijas.

Ten en cuenta que la popularidad es del año en que se publicó el juego y a medida que pasan los años se les resta 5 puntos.
Si la popularidad es menor a 0, queda con valor 0 y no negativo.
Asuma que no existen dos juegos con la misma popularidad y año.

Entrada

La capacidad total del disco duro del PS5 Slim.
Año actual.
La cantidad de juegos a recibir.
Una lista de juegos, donde cada juego está representado por una tupla (nombre en CamelCase, tamaño en GB, popularidad original, año de lanzamiento).

Salida

Popularidad máxima ajustada que puedes alcanzar.

Ej: si recibimos la siguiente lista de juegos:

FinalFantasyXV 40 90 2021
GodOfWar 20 85 2022
TheLastOfUs 30 80 2020
HorizonZeroDawn 50 70 2023

Y teniendo en cuenta el año actual 2023, su popularidad ajustada seria:

FinalFantasyXV: 90 - 5*(2023-2021) = 80
GodOfWar: 85 - 5*(2023-2022) = 80
TheLastOfUs: 80 - 5*(2023-2020) = 65
HorizonZeroDawn: 70 - 5*(2023-2023) = 70

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Restricciones

- **Orden Temporal y espacial: O(n.storage)** siendo n la cantidad de juegos y storage el almacenamiento maximo de la consola

Ejemplo de Entrada

100
2023
4
FinalFantasyXV 40 90 2021
GodOfWar 20 85 2022
TheLastOfUs 30 80 2020
HorizonZeroDawn 50 70 2023

Ejemplo de Salida

225

10. Sudoku

Nombre de archivo: ejercicio10.cpp/Ejercicio10.java

Descripción del problema

Un amigo experto en sudoku quiere hacer una competencia con usted en sudoku competitivo. Este sudoku competitivo tiene una variacion respecto al sudoku comun. Esta variacion es que tiene tableros de distintas dimenciones, como por ejemplo 9x9, 12x9, 9x12, etc. Dado que usted no es muy bueno en sudoku, pero no quiere perder contra su amigo decide hacer trampa. Para ello, hara un programa que dado un tablero de sudoku, le imprime la solucion del mismo.

Las reglas del sudoku son las siguientes:

- En una fila no puede haber dos numeros repetidos.
- En una columna no puede haber dos numeros repetidos.
- Si dividimos el tablero en 9 bloques (de <filas totales/3> filas * <columnas totales/3> columnas), en cada bloque no puede haber numeros repetidos. Y los numeros del 1 al k, sin saltearse ningun número.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Entrada

<filas> <columnas>
<fila 1>
...
<fila #columnas>

Salida

<fila 1>
...
<fila #columnas>

Restricciones

- **Hacerlo con Backtracking**

Ejemplo de Entrada

9 9
2 1 3 6 9 8 4 5 7
5 6 7 1 2 4 3 8 9
4 8 9 7 5 3 1 6 2
3 2 1 0 6 9 5 7 8
6 4 0 8 0 7 2 9 3
7 0 8 2 3 5 6 1 4
1 7 2 3 0 6 0 4 5
8 5 6 9 0 2 7 0 1
9 3 0 5 7 1 8 2 6

Ejemplo de Salida

2 1 3 6 9 8 4 5 7
5 6 7 1 2 4 3 8 9
4 8 9 7 5 3 1 6 2
3 2 1 4 6 9 5 7 8
6 4 5 8 1 7 2 9 3
7 9 8 2 3 5 6 1 4
1 7 2 3 8 6 9 4 5
8 5 6 9 4 2 7 3 1
9 3 4 5 7 1 8 2 6

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

RECORDATORIO: IMPORTANTE PARA LA ENTREGA

Obligatorios (Cap.IV.1, Doc. 220)

La entrega de los obligatorios será en formato digital online, a excepción de algunas materias que se entregarán en Bedelía y en ese caso recibirá información específica en el dictado de la misma.

Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. La entrega se realizará desde **gestion.ort.edu.uy**
2. Previo a la conformación de grupos cada estudiante deberá estar inscripto a la evaluación. S**ugerimos realizarlo con anticipación.**
3. **Uno de los integrantes del grupo de obligatorio será el administrador del mismo** y es quien formará el equipo y subirá la entrega
4. Cada equipo debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar)
5. El archivo a subir debe tener **un tamaño máximo de 40 mb**
6. Les sugerimos **realicen una 'prueba de subida' al menos un día antes**, donde conformarán el **'grupo de obligatorio'.**
7. La **hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
8. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc)
9. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor pasar por la oficina del Coordinador o por Coordinación adjunta **antes de las 20:00hs.** del día de la entrega

Si tuvieras una situación particular de fuerza mayor, debes dirigirte con suficiente antelación al plazo de entrega, al Coordinador de Cursos o Secretario Docente.