

# SEMANA 8 - Concurso de Prueba

---

Descripción de soluciones

# Problema A - Palabras Encadenadas

Envíos: **36**

Envíos Aceptados: **7**

% de acierto: **19.44%**

# Problema A - Palabras Encadenadas

La idea tras este problema era comparar los dos caracteres de inicio con todas las palabras que coincidieran con los dos últimos caracteres del final.

Una palabra no se comparaba consigo misma.

Al ser el problema con un  $N$  tan corto, bastaba con hacer un algoritmo bruto que comparara una palabra con todas las otras y luego haciendo un substring de esa palabra para compararla. Total  $O(N^3)$

# Problema B - Tendencias en Redes Sociales

Envíos: **18**

Envíos Aceptados: **6**

% de acierto: **33.33%**

## Problema B - Tendencias en Redes Sociales

Para este problema. Por cada nombre, evaluabamos directamente el array de enteros que nos daban, como siempre nos daban al menos dos valores, eso lo podíamos representar en un bucle de 1 hasta N.

Comprobamos que  $V[i] > V[i-1]$  para todo  $i$  de  $1..N$ , si era cierto, imprimíamos el nombre, si no, no imprimíamos nada.

# Problema C - Próximo Año Bisiesto

Envíos: **58**

Envíos Aceptados: **8**

% de acierto: **13.79%**

# Problema C - Próximo Año Bisiesto

Para este problema habían dos posibles acercamientos.

Iterar desde  $Y+1$  hasta infinito y luego comprobar si  $Y+1+i$  era bisiesto.

Devolverte al “último año divisible entre 4” y, como 100 y 400 también son divisibles entre 4, iterar de 4 en 4. Ej.

$1999 \% 4 \Rightarrow 3$ ,  $1999-3 = 1996$  (último año bisiesto, no nos interesa por ser  $<Y$  por lo que sumamos 4)  $\Rightarrow 1996+4 = 2000$  (es un año bisiesto)

La condición de un año bisiesto es si  $Y\%4 == 0 \ \&\& \ (Y\%100 != 0 \ || \ Y\%400 == 0)$

# Problema D - Agencia de Espías

Envíos: **7**

Envíos Aceptados: **1**

% de acierto: **14.29%**



# Problema D - Agencia de Espías

Problema sobre grafos. Problema clásico del grafo bipartito.

Dado un nodo origen, arbitrariamente 'pintarlo' de un color (llevar dos arrays; uno de visitados y otro del color que tiene cada nodo) y hacer un BFS / DFS sobre él, teniendo en consideración que si algún nodo visitado está coloreado, se evaluará si los colores son iguales (y si lo son, el grafo NO es bipartito).

Había que tener especial cuidado con las componentes conexas (el grafo podía estar completamente aislado)

# Problema E - Versiones de Programas

Envíos: **27**

Envíos Aceptados: **1**

% de acierto: **3.70%**

# Problema E - Versiones de Programas

Un problema clásico para entender la funcionalidad de Map/Filter/Reduce para los lenguajes que las posean (Python y Java8).

Separar en un array todos los números contenidos en el string.

Parsear a enteros todos esos números y luego ordenarlos para todo  $i$  si  $X[i] == Y[i]$  para todo el bucle se ordenaba por quien tuviera mayor tamaño. Si en algún momento del bucle  $X[i] > Y[i]$ ;  $X$  era mayor, y si  $X[i] < Y[i]$ ;  $Y$  era mayor.

El resultado lo pasabas del array de números a un string separados por un ‘.’

# RESULTADOS

EQUIPO	A	B	C	D	E	SCORE	PENALTY
Victorious Secret	56(+1)	50(+2)	25(+1)	98(+1)	86(+10)	5	615
RaspuTeam	54	48(+4)	10(+1)	+5	+5	3	212
Peps	86	113	18		+1	3	217
JoseFco	33(+4)	84(+1)	58(+1)			3	295
P3450	116(+1)	103	55(+1)		+2	3	314
Stanislav	41		59(+1)		+2	2	120
Rocódromos Arguelles 2	48(+1)		34			2	162
Rocódromos Arguelles 1	+6	83(+1)	+8			1	103
Adrián			53(+3)			1	113