



Concurso Final 2023

Estadísticas y Soluciones



Clasificación de los problemas

Problema	Categoría
A - Otro aburrido problema de Balonmano	Pensar
B - ¿Van tías?	Pensar
C - Fenómenos	Grafos
D - La ladrona de libros	Pilas y Colas. Time Waster.
E - Fuga dall'ascensore	Programación Dinámica
F - ¡Otra vez!	Mapas
G - URJC Dates	Ordenación

Estadísticas

Problema	# casos de prueba	Espacio en disco
A - Otro aburrido problema de balonmano	4	2.7 MB
B - ¿Van tías?	6	980 KB
C - Fenómenos	5	2.66 MB
D - La ladrona de Libros	11	1.1 MB
E - Fuga dall'ascensore	25	30.2 MB
F - ¡Otra vez!	57	12.4 MB
G - URJC Dates	6	231 KB
- Total	109	55.6 MB (+-)

Estadísticas*

Problema	Primer equipo en resolverlo	Tiempo
A - - Otro aburrido problema de Balonmano	AColorados	7
B - - ¿Van tias?	MIIDAS AC?	61
C - - Fenómenos	AColorados	36
D - - La ladrona de libros	¿?	¿?
E - - Fuga dall'ascensore	¿?	¿?
F - - ¡Otra vez!	MIIDAS AC?	21
G - - URJC Dates	C-gala	28

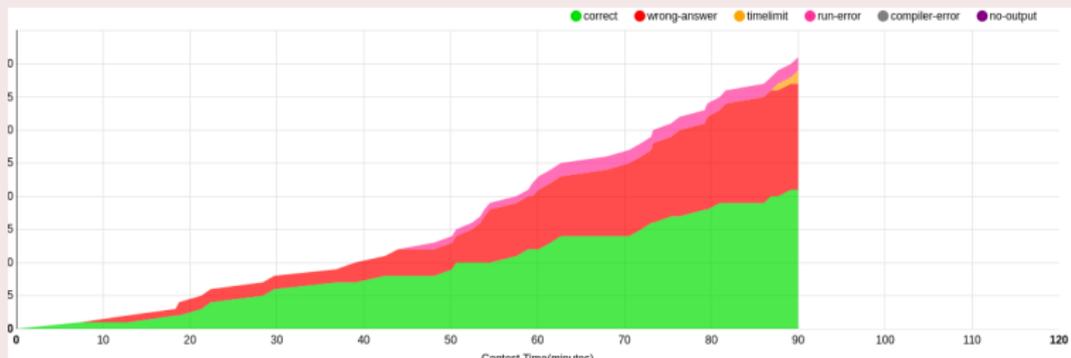
* Antes de congelar el marcador.

Estadísticas*

Problema	Envíos	Válidos	% éxito
A - Otro aburrido problema de Balonmano	17	9	53 %
B - ¿Van tías?	2	1	50 %
C - Fenómenos	1	1	100 %
D - La ladrona de libros	0	0	0 %
E - Fuga dall'ascensore	2	0	100 %
F - ¡Otra vez!	9	5	56 %
G - URJC Dates	11	5	45 %

* Antes de congelar el marcador.

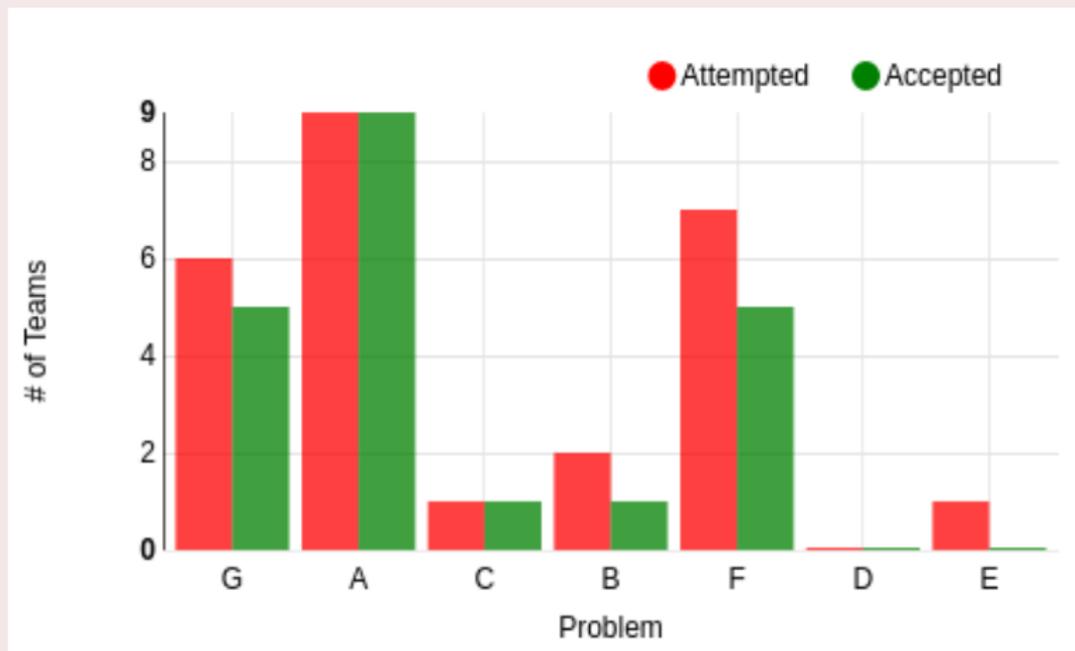
Estadísticas varias



Estadísticas varias



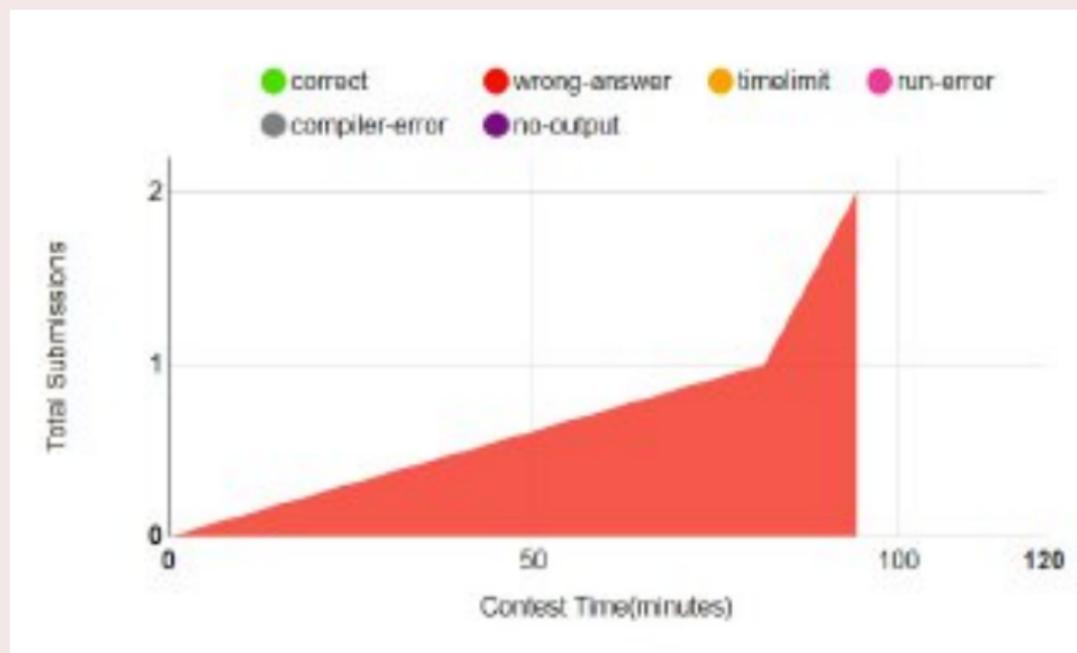
Estadísticas varias



● E. Fuga dall'ascensore

Envíos	Válidos	% éxito
2	0	0%

E. Fuga dall'ascensore



E. Fuga dall'ascensore

Dado un arreglo de N números enteros, ¿cuántas combinaciones de M elementos en el arreglo suman una cantidad objetivo S ?

E. Fuga dall'ascensore

Complejidad

Probar todas las combinaciones ? $\rightarrow N! / ((K!) * (N-K)!)$ BT

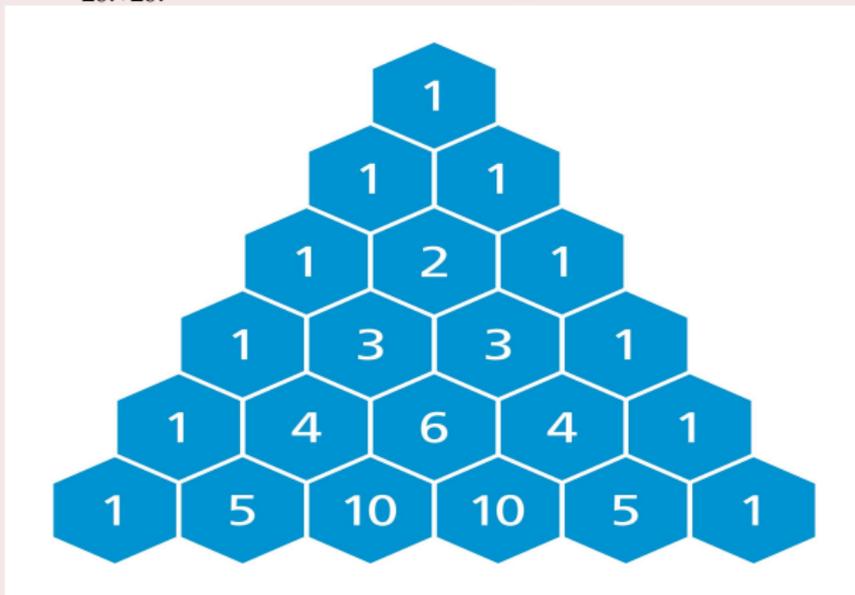
DP N^3

<u>input size</u>	<u>required time complexity</u>
$n \leq 10$	$O(n!)$
$n \leq 20$	$O(2^n)$
$n \leq 500$	$O(n^3)$
$n \leq 5000$	$O(n^2)$
$n \leq 10^6$	$O(n \log n)$ or $O(n)$
n is large	$O(1)$ or $O(\log n)$

E. Fuga dall'ascensore

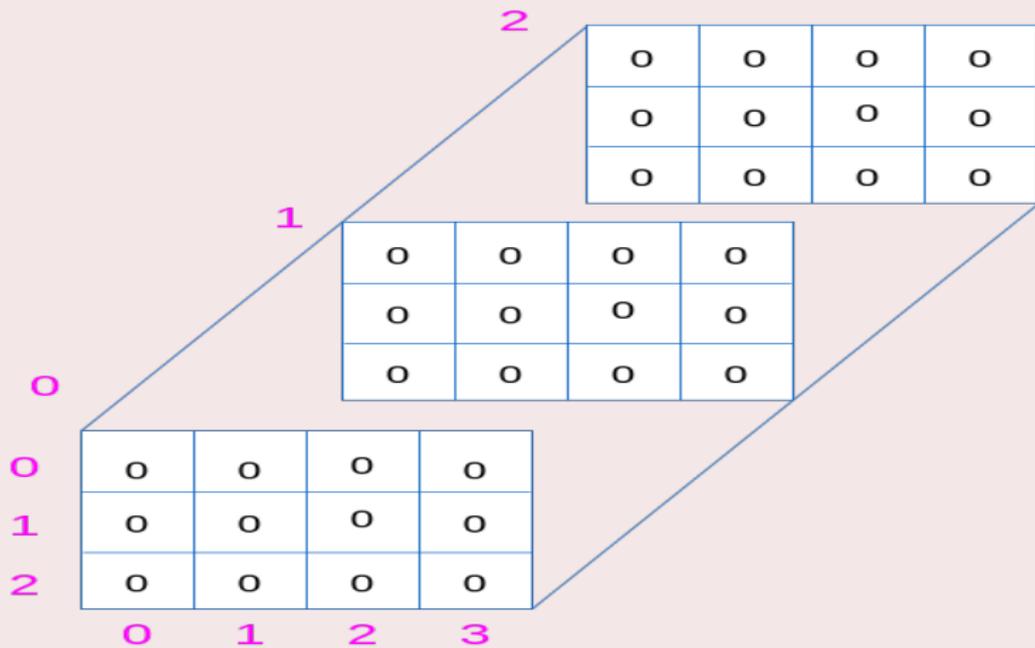
Parte tricky Long Long

$$\frac{40!}{20!*20!} = 137846528820 \text{ MAX_INT}=4294967296$$

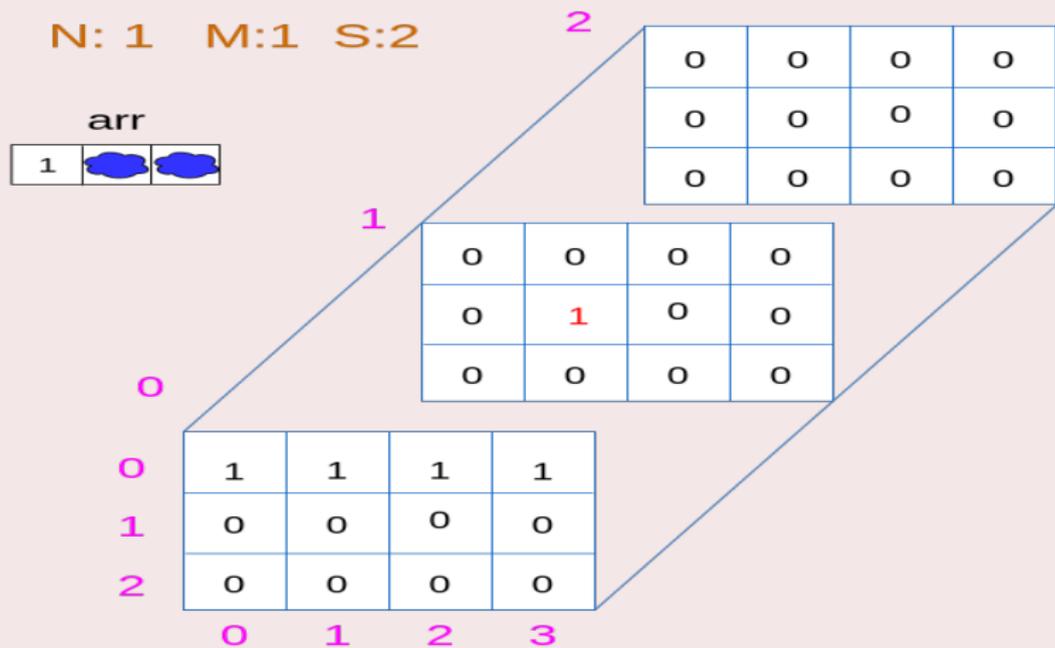


E. Fuga dall'ascensore

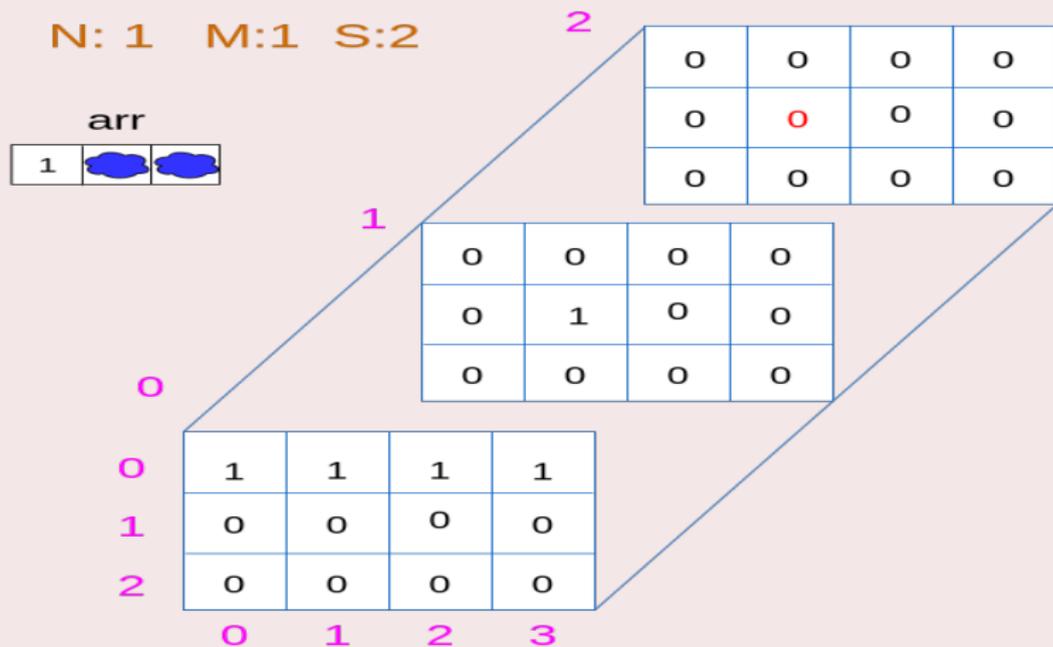
1
3 2 2
1 1 1



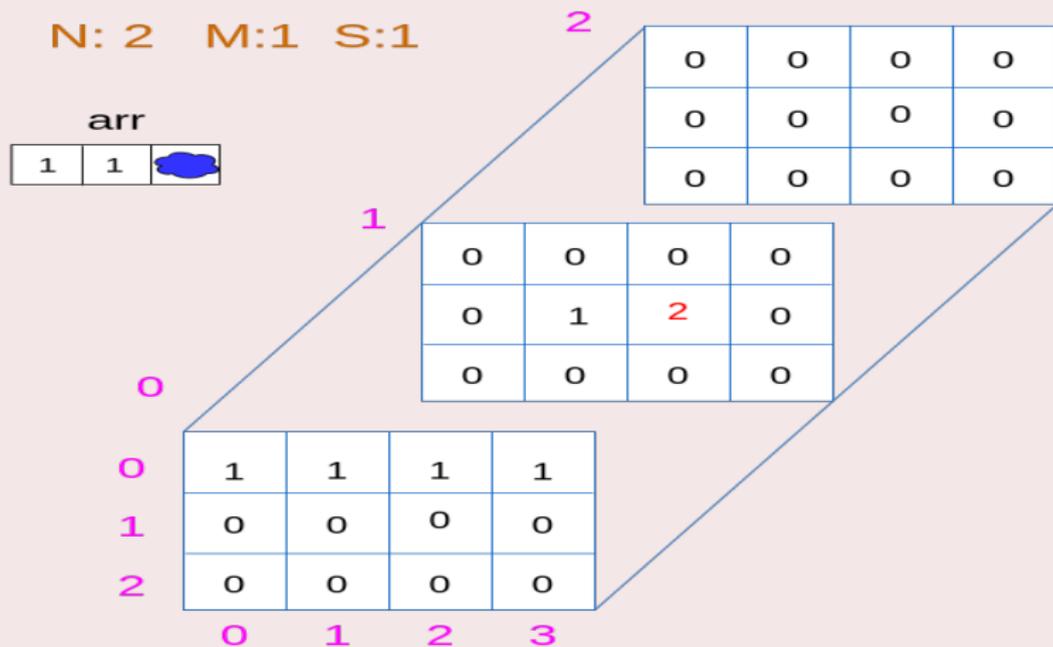
E. Fuga dall'ascensore



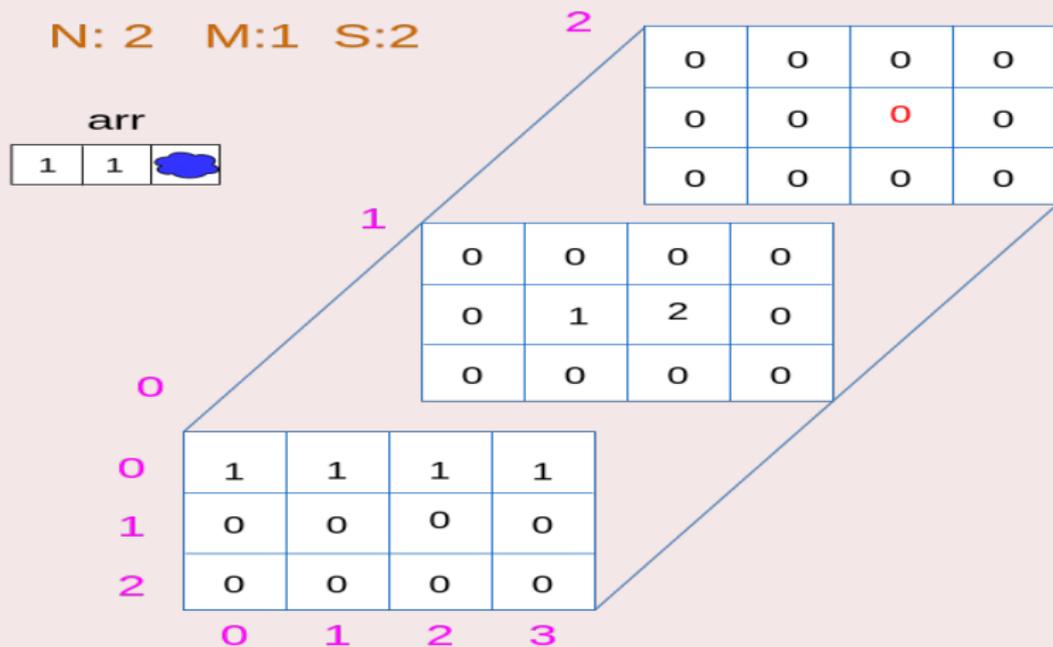
E. Fuga dall'ascensore



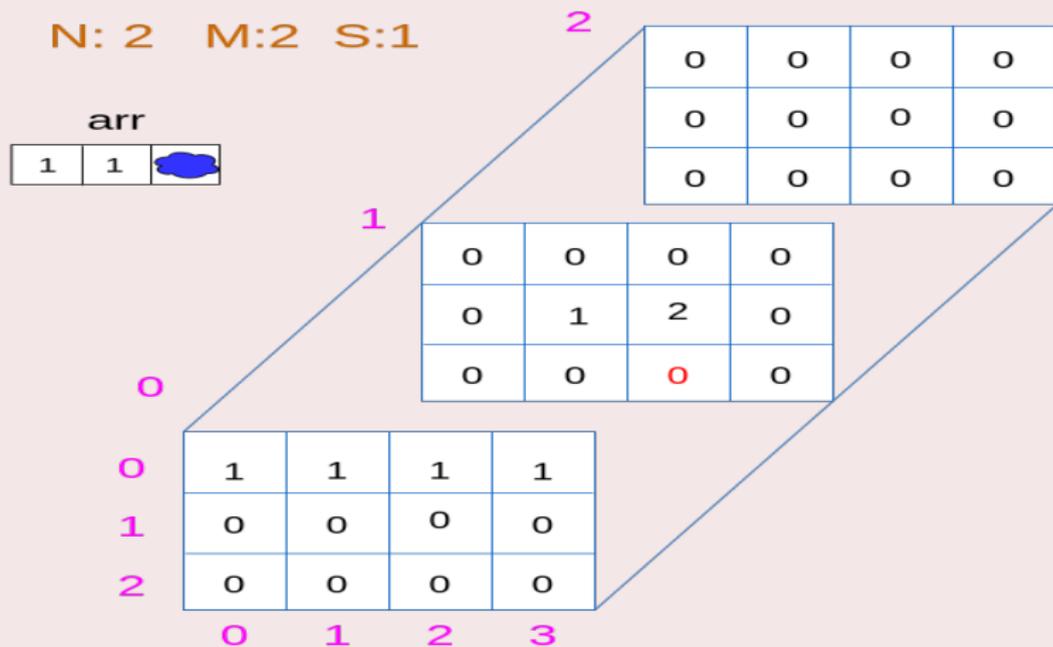
E. Fuga dall'ascensore



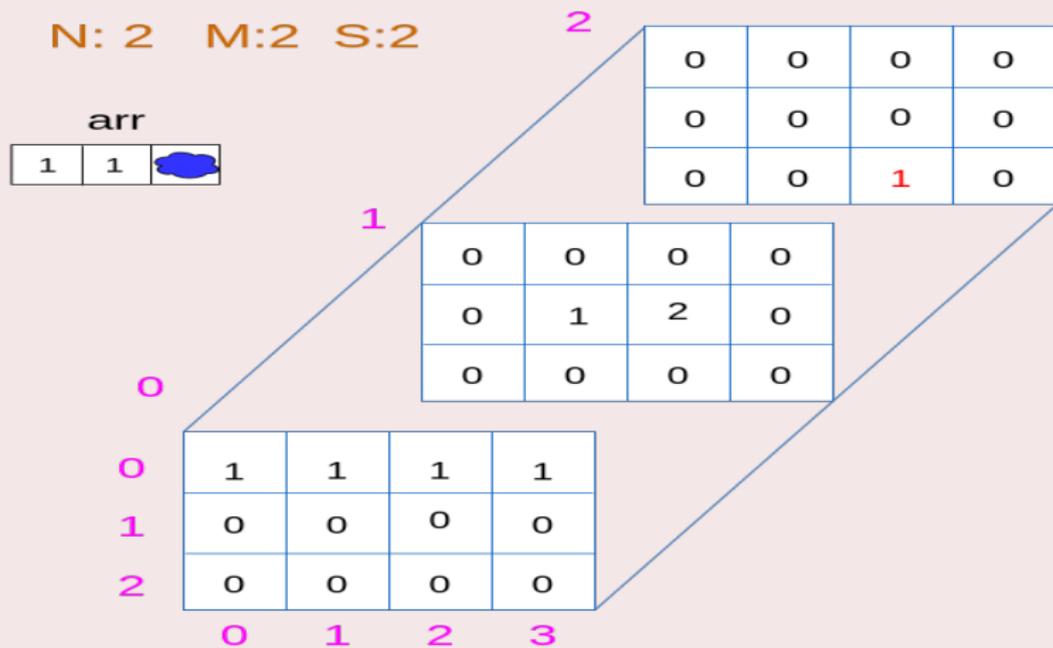
E. Fuga dall'ascensore



E. Fuga dall'ascensore



E. Fuga dall'ascensore

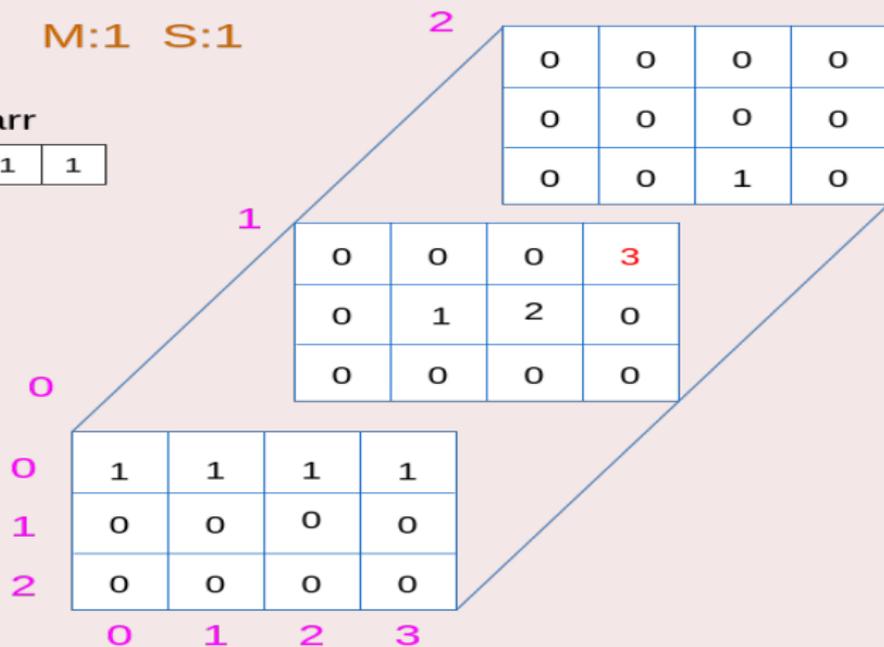


E. Fuga dall'ascensore

N: 3 M:1 S:1

arr

1	1	1
---	---	---

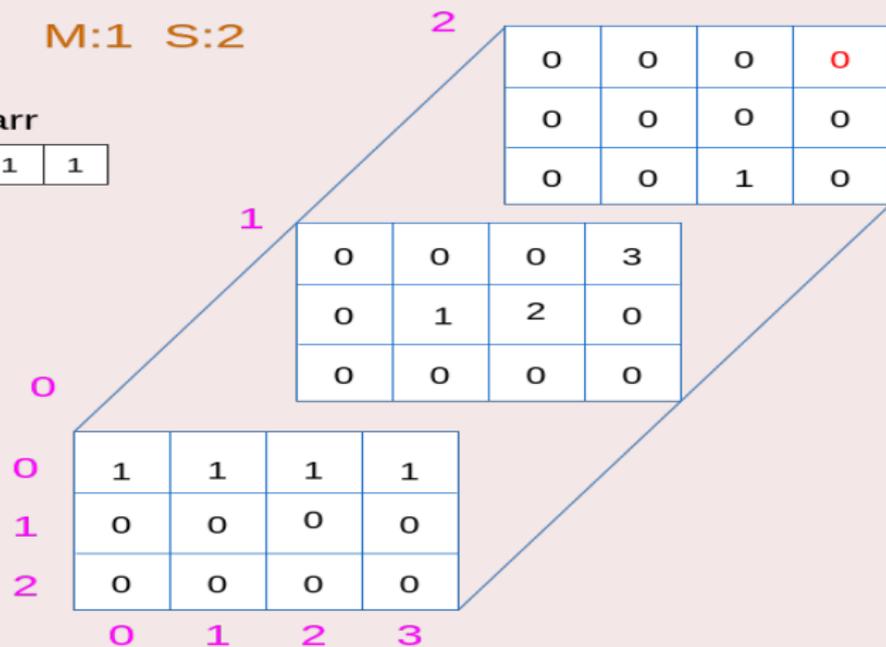


E. Fuga dall'ascensore

N: 3 M:1 S:2

arr

1	1	1
---	---	---

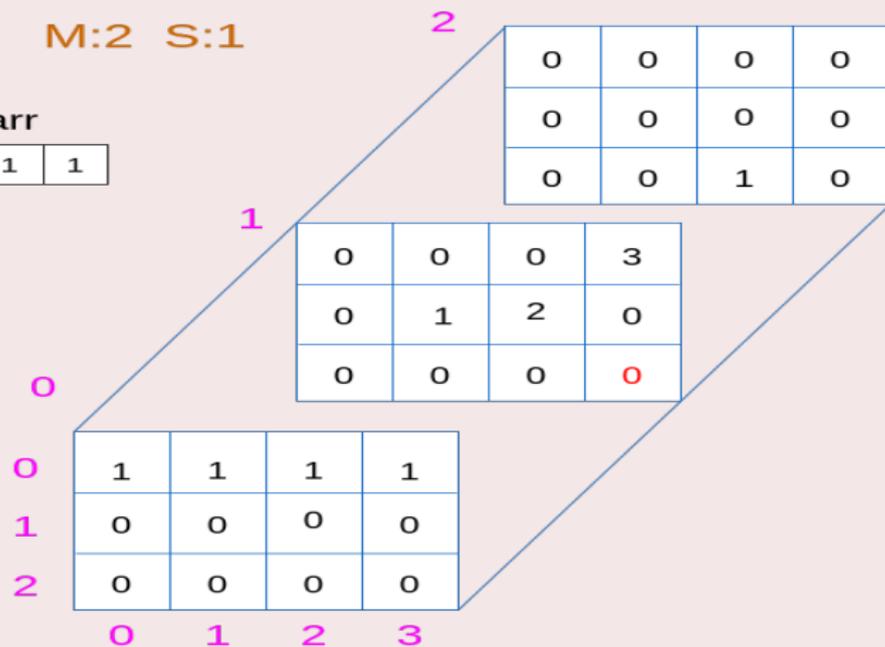


E. Fuga dall'ascensore

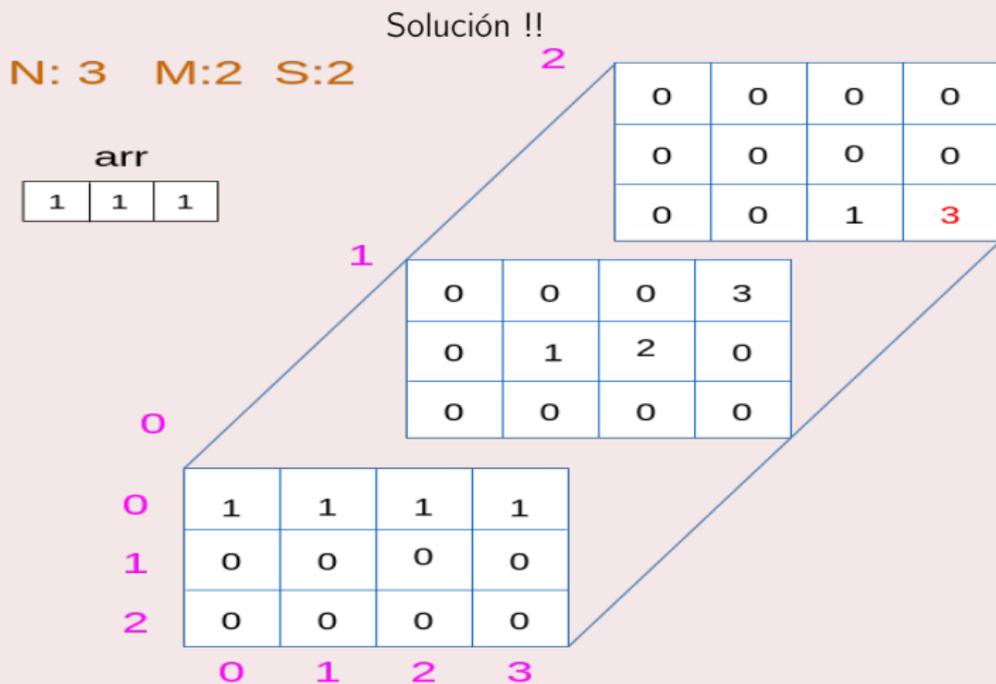
N: 3 M:2 S:1

arr

1	1	1
---	---	---



E. Fuga dall'ascensore



E. Fuga dall'ascensore

```

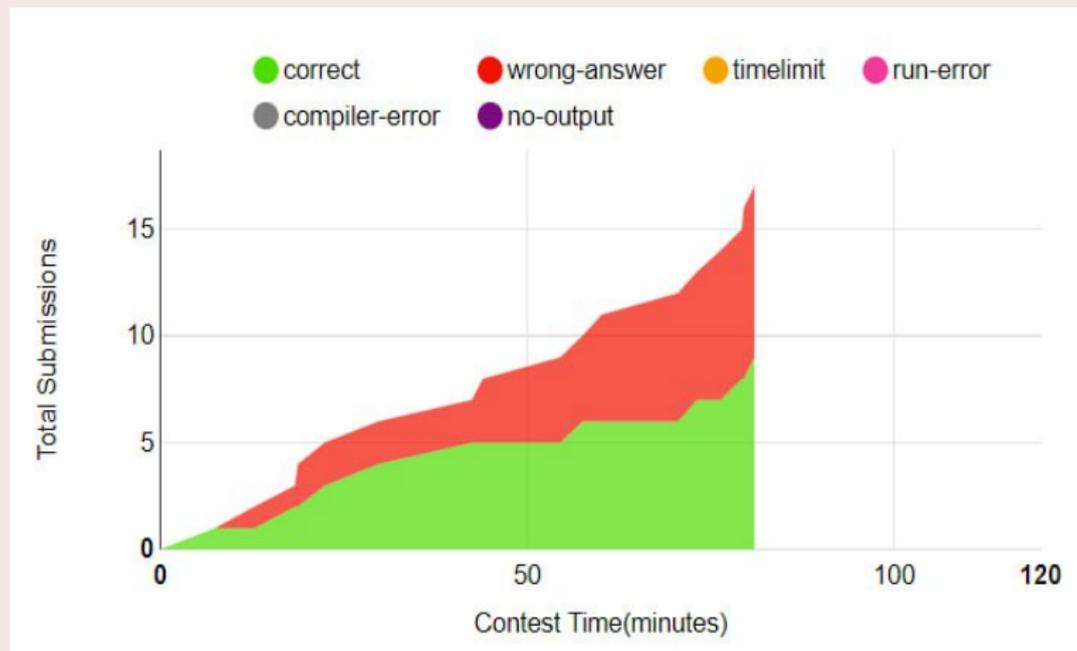
memset(dp,0, sizeof(dp));
for (ll i =0; i <=n; ++i) {dp[i][0][0]=1;}
for (ll i = 1; i <=n ; ++i) {
    for (ll j = 1; j<=i && j<=k ; ++j) {
        for (ll l = 1; l <=target; ++l) {
            dp[i][j][l]=dp[i-1][j][l];
            if(l>=arr[i]){
                dp[i][j][l]+=dp[i-1][j-1][l-arr[i]];
            }
        }
    }
}
}

```

● A. Otro aburrido problema de
Balonmano

Envíos	Válidos	% éxito
17	9	53 %

A. Otro aburrido problema de Balonmano



A. Otro aburrido problema de Balonmano

El problema fácil del concurso!!! Solo teníamos que pensar un poco... El problema simulaba los resultados de una competición deportiva:

CLASIFICACIÓN

		J	G	E	P	GF	GC	DIF	PUNTOS
1	 CORE GLOBAL BALONMANO PARLA	16	16	0	0	646	444	202	32
2	 QUENTAL BM PINTO	16	12	0	4	506	420	86	24
3	 VIRGEN DE EUROPA	16	7	0	9	483	484	-1	14
4	 RECUERDO 3CM	16	5	0	11	464	538	-74	10

A. Otro aburrido problema de Balonmano

Había que identificar 3 pistas que hacían el problema trivial:

- Cada vez que se presentaba la clasificación de una jornada, se hacía en orden de posición. Es decir, el primero de la lista es el primero del torneo.
- Las jornadas aparecen en orden inverso al temporal. Es decir, la primera clasificación corresponde al último día del torneo, AL FINAL DEL TORNEO.
- Los nombres de los equipos eran una sola palabra.

Con estas ideas solo había que imprimir el nombre del equipo en primer lugar en la primera clasificación del input: LA PRIMERA PALABRA DE LA CUARTA LINEA.

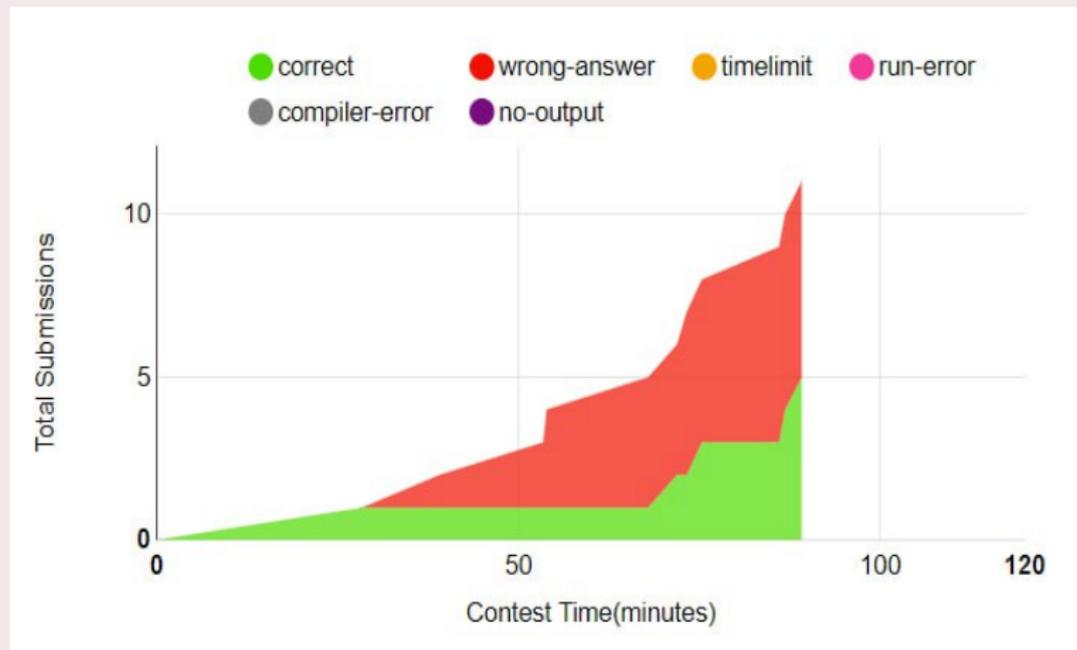
A. Otro aburrido problema de Balonmano

```
n = int(input())
input()
input()
line = input()
pf = line.split(" ")
ganador = pf[0]
print(ganador)
```

● G. URJC Dates

Envíos	Válidos	% éxito
11	5	45 %

G. URJC Dates



G. URJC Dates

El problema nos habla de dos grupos de personas donde cada persona tiene asociado un número que denota su belleza. Nos pedía obtener la menor penalización (minimizar la suma de la diferencia entre los números a la hora de emparejar). ¿Solución?

G. URJC Dates

Lo primero era darse cuenta de que el nombre de las personas no aportaba nada.

Por lo tanto el problema se reduce a dos arrays de enteros.

Una vez nos damos cuenta de esto, basta con ordenar los dos arrays e ir restando los números que están en la misma posición (en valor absoluto).

G. URJC Dates

¿Por qué esto se cumple? La respuesta es porque estamos juntando los números de un grupo con el número que más se le parece del otro grupo. Y de esta forma estamos consiguiendo que la diferencia sea mínima en cada caso y por tanto la suma sea la menor posible.

G. URJC Dates

Fallo general: Veamos el siguiente caso:

$$a_1 = \{10, 8\}$$

$$a_2 = \{9, 9\}$$

Si sumamos los dos arrays y restamos obtenemos lo siguiente: $|18 - 18| = 0$

Sin embargo la solución real es $|10 - 9| + |8 - 9| = 2$

G. URJC Dates

```
leer(N)
for(i=0;i<Ni++)
    leer(nombre)
    leer(puntuacion)
    grupo1.add(puntuacion)
for(i=0;i<Ni++)
    leer(nombre)
    leer(puntuacion)
    grupo2.add(puntuacion)

sort(grupo1)
sort(grupo2)

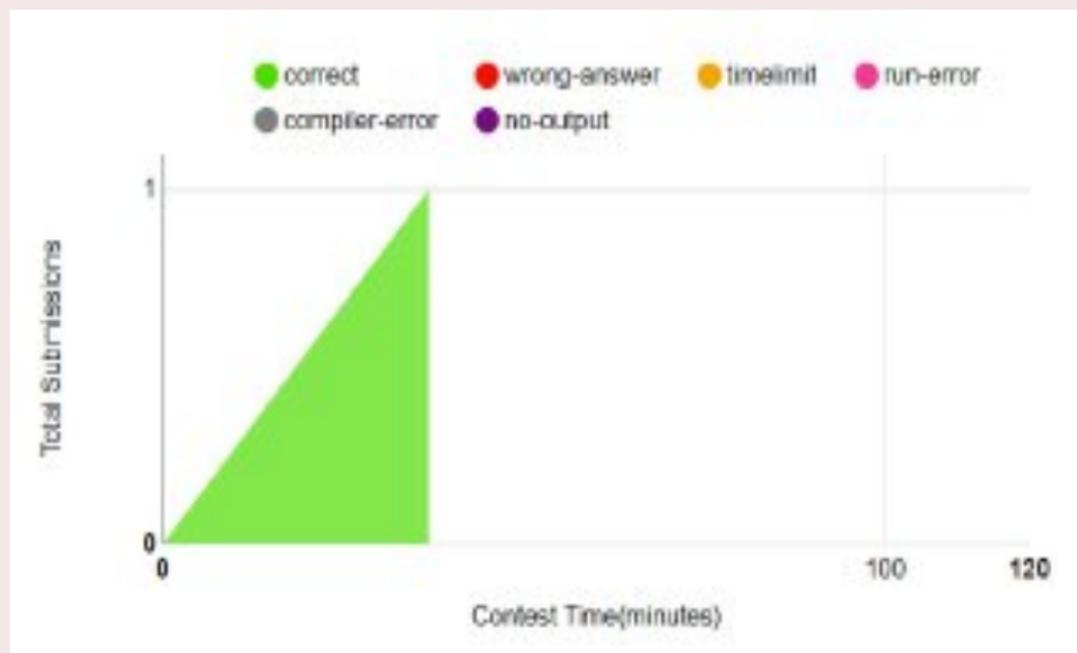
int sol = 0
for(i=0;i<N;i++)
    sol += abs(grupo1[i]-grupo2[i])

imprimir(sol)
```

● C. Fenómenos

Envíos	Válidos	% éxito
1	1	100 %

C. Fenómenos



C. Fenómenos

AüLA 3



AüLA 5



AüLA 4



AüLA 0

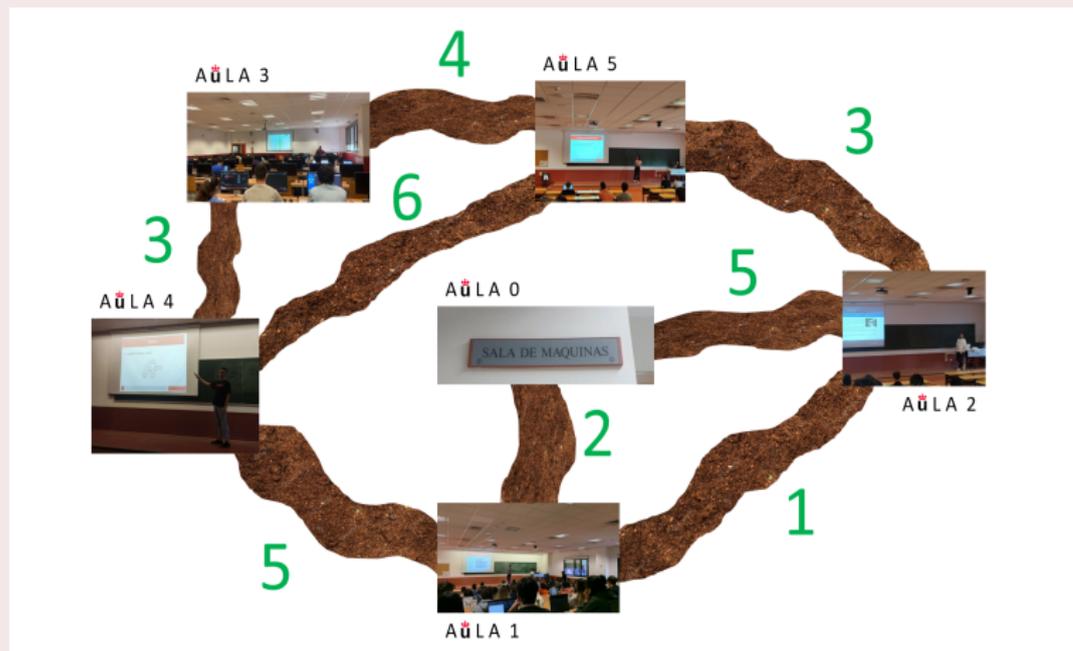


AüLA 2



AüLA 1

C. Fenómenos



C. Fenómenos



C. Fenómenos

¿Máximo ahorro?

C. Fenómenos

Marta
Sebas
Tú

C. Fenómenos

Minimum
Spanning
Tree

C. Fenómenos



C. Fenómenos

$$\begin{aligned} &\text{Máximo ahorro} \\ &= \\ &\text{Coste total} - \text{costeMST} \end{aligned}$$

C. Fenómenos

```
leer C
```

```
para cada C:
```

```
    leer a, t
```

```
    leer aristas
```

```
    calcular costeTotal
```

```
    //Algoritmo para calcular el costeMST
```

```
    costeMST = 0
```

```
    hasta grafo conexo:
```

```
        coger arista de menor coste que una un nodo nuevo
```

```
        conectar nodo al grafo
```

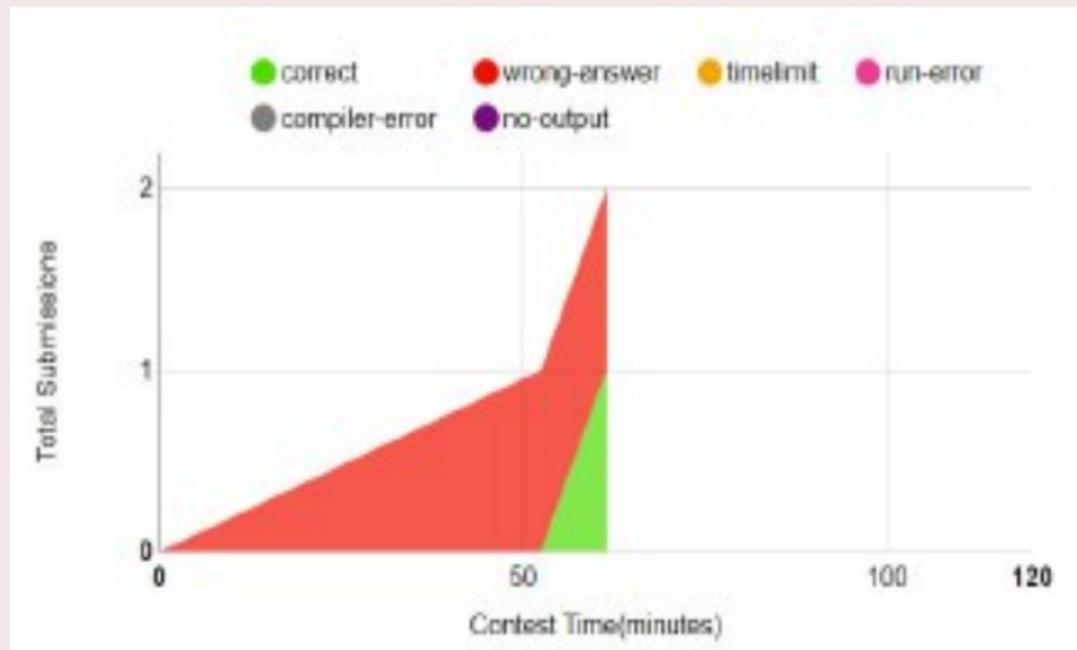
```
        sumar coste de la arista al costeMST
```

```
    imprimir costeTotal-costeMST
```

● B. ¿Van tias?

Envíos	Válidos	% éxito
2	1	50 %

B. ¿Van tias?



B. ¿Van tias?

Dado un laberinto cuya entrada se encuentra en el muro inferior y la salida en el muro superior, y para el cual se garantiza que existe un camino que llega a la salida, ¿se podrá salir utilizando la técnica de la mano derecha?

B. ¿Van tias?

-Posible idea:

B. ¿Van tias?

-Posible idea:

Simular el recorrido de la mano derecha empezando por la entrada, comprobando si se llega a la salida.

B. ¿Van tias?

-Posible idea:

Simular el recorrido de la mano derecha empezando por la entrada, comprobando si se llega a la salida.

-Mucho más fácil que eso...

B. ¿Van tias?

¿Qué puede ocurrir si seguimos la técnica de la mano derecha?

B. ¿Van tias?

¿Qué puede ocurrir si seguimos la técnica de la mano derecha?

A) Volvemos a la entrada sin pasar por la salida

B. ¿Van tias?

¿Qué puede ocurrir si seguimos la técnica de la mano derecha?

- A) Volvemos a la entrada sin pasar por la salida
- B) **Llegamos a la salida**

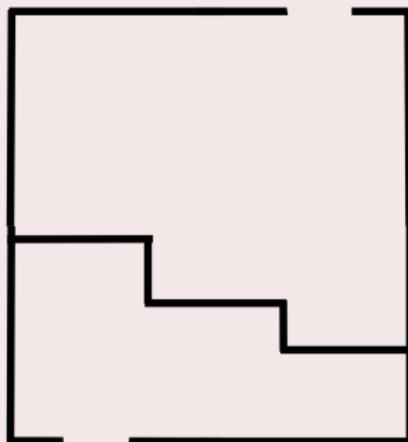
B. ¿Van tias?

¿En qué casos ocurre A) y NO saldremos del laberinto?

B. ¿Van tias?

¿En qué casos ocurre A y NO saldremos del laberinto?

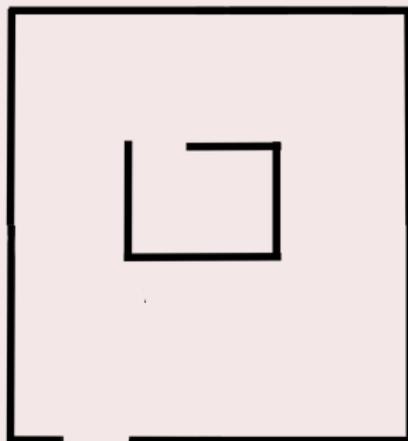
1. No existe un camino a la salida, quedamos bloqueados por los muros del laberinto.



B. ¿Van tias?

¿En qué casos ocurre A y NO saldremos del laberinto?

1. No existe un camino a la salida, quedamos bloqueados por los muros del laberinto.
2. La entrada/ salida se encuentra en una isla desconectada del resto del laberinto



B. ¿Van tias?

¿En qué casos ocurre A y NO saldremos del laberinto?

1. **IMPOSIBLE**: existe un camino hasta la salida
2. **IMPOSIBLE**: la entrada y salida se encuentran en la misma “isla”: el muro exterior del laberinto

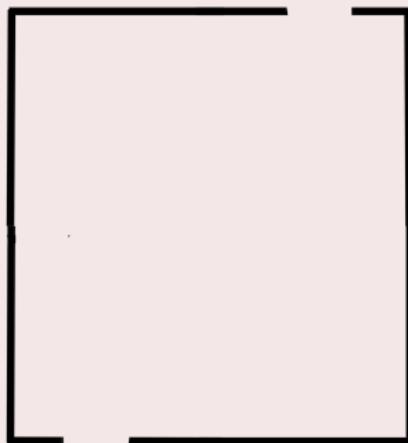
B. ¿Van tias?

En definitiva, con estas condiciones **SIEMPRE** se podrá salir.

B. ¿Van tias?

En definitiva, en estas condiciones **SIEMPRE** se podrá salir.

Se ve fácil de manera visual, puesto que los laberintos descritos en el enunciado son de la forma:



B. ¿Van tias?

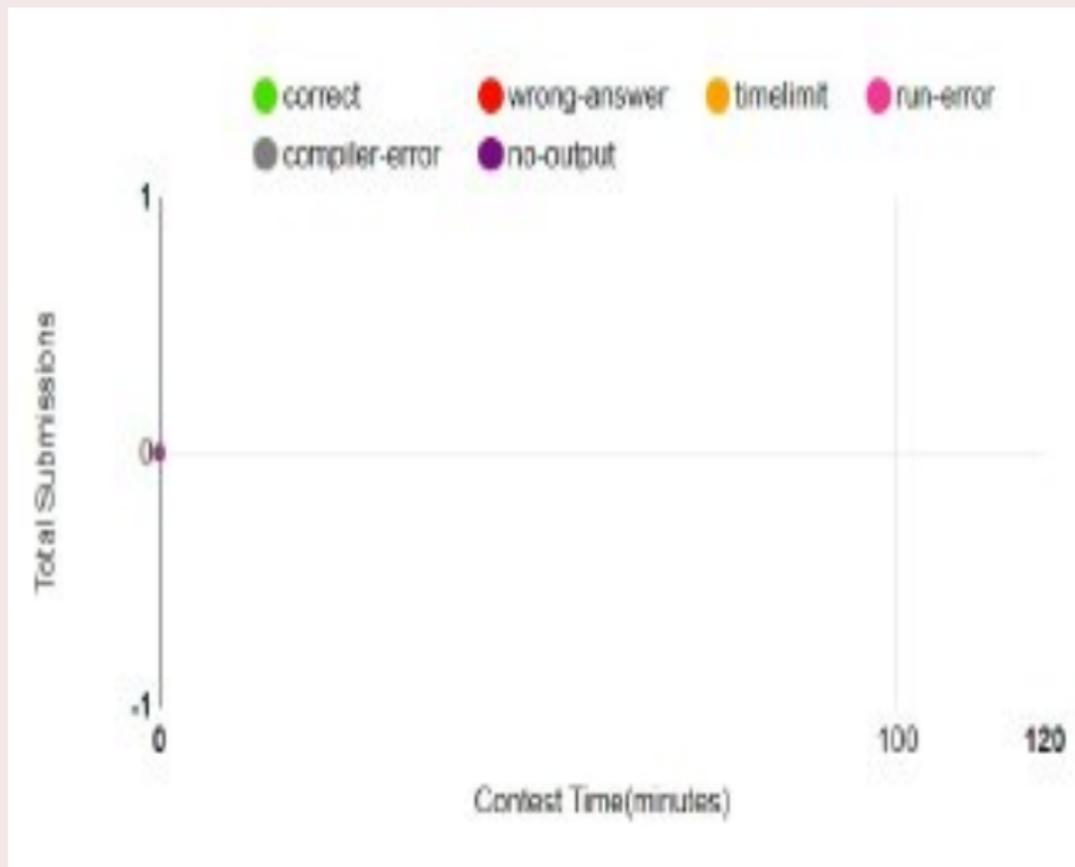
Pseudocódigo:

```
\\por cada caso de prueba  
print("Hoy se sale :)")
```

● D. La ladrona de libros

Envíos	Válidos	% éxito
0	0	0%

D. La ladrona de libros



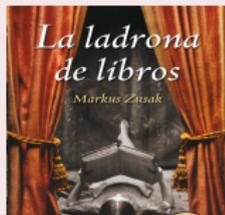
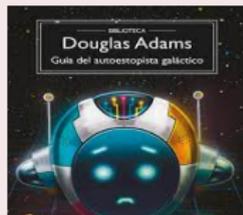
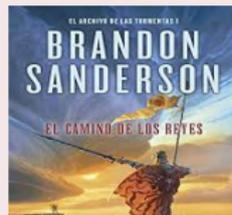
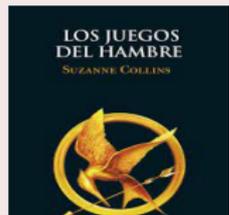
D. La ladrona de libros

Simplemente debemos guardar toda la información de la tienda de manera que podamos simular los eventos de manera eficiente:

D. La ladrona de libros

Simplemente debemos guardar toda la información de la tienda de manera que podamos simular los eventos de manera eficiente:

ESTANTERIA

**1****2****3****4**

D. La ladrona de libros

Simplemente debemos guardar toda la información de la tienda de manera que podamos simular los eventos de manera eficiente:

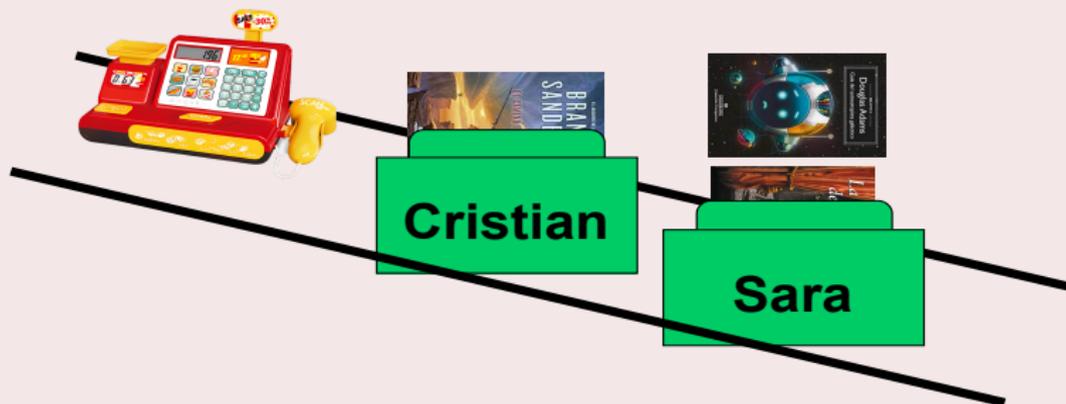
CESTAS



D. La ladrona de libros

Simplemente debemos guardar toda la información de la tienda de manera que podamos simular los eventos de manera eficiente:

CAJA



D. La ladrona de libros

- Estantería: Array o Mapa
- Cestas de los clientes: Pilas ("Las cestas de tu tienda son demasiado estrechas por lo que si un cliente coge varios libros deberá apilarlos de forma que solo podrá acceder al último libro que ha cogido.")
- Caja registradora: Cola de cestas = Cola de Pilas! ("Tú recibirás los libros en la caja para cobrarlos, de esa manera si no se registran en el sistema de la caja en el orden natural, es que el muy desgraciado se ha guardado uno en algún lado.")

D. La ladrona de libros

- Evento 1: Introducir en la pila el libro indicado : **cesta[cliente].push(libro)**
- Evento 2: Eliminar en la pila el libro indicado: **cesta[cliente].pop()**
- Evento 3: Añadir cesta a la cola: **cola.push(cesta[cliente])**

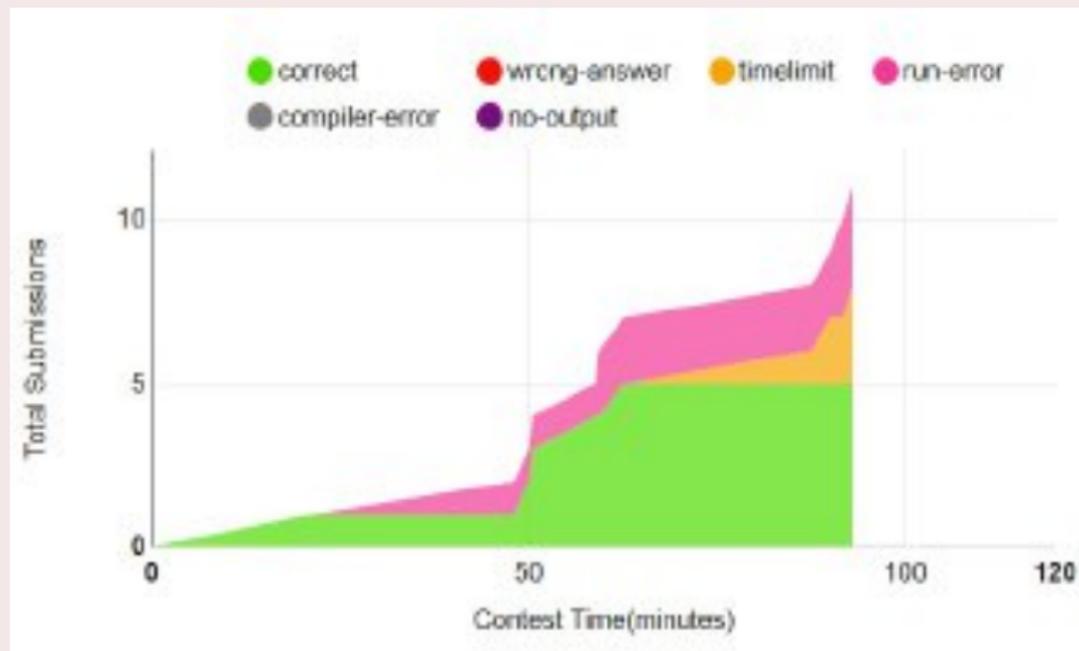
D. La ladrona de libros

Una vez se han procesado todos los eventos, se debe comprobar la lista de la entrada, con la de la cola. En caso de que no coincida es porque falta un libro. La cesta del cliente que estemos revisando en ese momento es la ladrona!!! Si todo coincide es que no ha venido la ladrona y no te está robando el poco dinero que ganas con tu tiendecita. OJO!!: Cuidado con el caso de que el último libro de la caja sea el robado, en ese caso todo coincidirá, pero la cesta registrada por los eventos no estará vacía!!

● F. ¡Otra Vez!

Envíos	Válidos	% éxito
9	5	56 %

F. ¡Otra Vez!



F. ¡Otra Vez!

Tián ha vuelto a liarla y ha desordenado los papeles de la asociación, y es tu tarea ayudarnos a recogerlos. El problema te da f ficheros identificados por dos cadenas de caracteres ID y N , el nombre del grupo al que pertenece dicho fichero y el nombre del propio fichero.

Dado g nombres de grupo, ¿puedes decirnos cuáles son los nombres de los documentos que pertenecen a cada uno?

F. ¡Otra Vez!

Posibles soluciones:

- Lista de pares (ID, N) , por cada nombre de grupo ID que me pidieran, iteraba por la lista y cogía todos los nombres de fichero e imprimía. $\mathcal{O}(f \cdot q)$, que en el caso límite $f = 1000000$ y $q = 1000$ da TLE.
- Crear un mapa donde las claves son los nombres de los grupos y los valores listas de nombres. Asumiendo que cada lista de nombres asociados a un grupo tiene un valor constante, la complejidad viene en las peticiones q . Esto nos dejaría con una complejidad $\mathcal{O}(q)$, que da AC.

F. ¡Otra Vez!

```
leer(n)
mapa = {}
para i en 0..n
    leer(grupo)
    leer(nombre)
    mapa[grupo].añadir(nombre)
leer(\n)
leer(q)
mientras q != "---"
    para s en mapa[q]
        imprimir(s)
    imprimir(\n)
    leer(q)
```