AdaByron 2017

Estadísticas y Soluciones

Clasificación de los problemas

Problema	Categoría
A - Aritmética verbal	recursión, vuelta atrás
B - Pixel Art	bucles, condicionales, cadenas
C - El profesor de música	grafos, caminos mínimos, fuerza bruta
D - Amigo invisible	arrays, grafos, permutaciones, pensar
E - El entretenimiento de las máquinas	cadenas, diccionarios, tries
F - Hijos a tope	árboles binarios, recursión
G - Chocolate con almendras	programación dinámica
H - Las partituras de la orquesta	búsqueda binaria, colas de prioridad
I - ¿Es múltiplo de 3?	condicionales, números triangulares, pensar
J - Más listos que el hambre	programación dinámica, grafos, caminos mínimos
K - Las perlas de la condesa	arrays, ordenación
L - Durmiendo en albergues	arrays, recorridos, bucles, cadenas

Estadísticas

Problema		# casos de prueba	Espacio en disco
A -	Aritmética verbal	6.096	659 KB
B -	Pixel Art	2.015	2 MB
C -	El profesor de música	200	716 KB
D -	Amigo invisible	20.586	597 KB
E -	El entretenimiento de las máquinas	1.634	2,5 MB
F -	Hijos a tope	24.182	848 KB
G -	Chocolate con almendras	8.207	295 KB
H -	Las partituras de la orquesta	8.199	2,6 MB
I -	¿Es múltiplo de 3?	15.103	159 KB
J -	Más listos que el hambre	6.778	296 KB
K -	Las perlas de la condesa	3.047	1,2 MB
L -	Durmiendo en albergues	2.335	9,6 MB
-	**Total**	98.382	21,5 MB

Estadísticas*

Problema	Primer equipo en resolverlo	Tiempo
A - Aritmética verbal	-	-
B - Pixel Art	Monos infinitos	12
C - El profesor de música	-	-
D - Amigo invisible	Monos infinitos	120
E - El entretenimiento de las máquinas	-	-
F - Hijos a tope	Team Burton	83
G - Chocolate con almendras	Monos infinitos	175
H - Las partituras de la orquesta	Team Burton	55
I - ¿Es múltiplo de 3?	Monos infinitos	4
J - Más listos que el hambre	-	-
K - Las perlas de la condesa	Monos infinitos	19
L - Durmiendo en albergues	Monos infinitos	29

*Antes de congelar el marcador.

Estadísticas*

Problema		Envíos	Válidos	% éxito
A -	Aritmética verbal	2	0	0 %
B -	Pixel Art	54	35	64 %
C -	El profesor de música	-	-	-
D -	Amigo invisible	32	6	18 %
E -	El entretenimiento de las máquinas	-	-	-
F -	Hijos a tope	11	7	63 %
G -	Chocolate con almendras	2	1	50 %
H -	Las partituras de la orquesta	56	5	8 %
I -	¿Es múltiplo de 3?	101	26	25 %
J -	Más listos que el hambre	7	0	0 %
K -	Las perlas de la condesa	60	24	40 %
L -	Durmiendo en albergues	68	23	33 %

*Antes de congelar el marcador.

Estadísticas*

	Jueces		Participantes	
Problema	**LOC**	**LOCNC**	**LOC**	**LOCNC**
A - Aritmética verbal	306,33	155,66	0	0
B - Pixel Art	83,66	51,00	71.37	56.91
C - El profesor de música	329,33	197,00	-	-
D - Amigo invisible	163,25	56,50	67.00	57.16
E - El entreten. de las máquinas	198,80	115,20	-	-
F - Hijos a tope	82,33	37,66	57.71	43.42
G - Chocolate con almendras	119,60	69,80	54.00	45.00
H - Las partituras de la orquesta	96,60	46,60	47.80	37.00
I - ¿Es múltiplo de 3?	75,50	26,00	38.80	25.00
J - Más listos que el hambre	190,40	77,20	0	0
K - Las perlas de la condesa	89,33	50,33	84.45	63.04
L - Durmiendo en albergues	85,40	42,40	56.86	44.00
- **Total**	1820,53	925,34		

*Antes de congelar el marcador.

```
123456789101112131415
161718192021222324252
627282930313233343536
373839404142434445464
748495051525354555657
585960616263646566676
869707172737475767778
798081828384858687888
990919293949596979899
```

I. ¿Es múltiplo de 3?

Envíos	Válidos	% éxito
101	26	25 %

I. ¿Es múltiplo de 3?

Recordatorio del enunciado

Dado un n, construir el número formado por la concatenación de los dígitos de los números $1 \dots n$, y decir si el resultado es múltiplo de 3: lo será si lo es la suma de sus dígitos.

I. ¿Es múltiplo de 3?

Recordatorio del enunciado

Dado un n, construir el número formado por la concatenación de los dígitos de los números $1 \ldots n$, y decir si el resultado es múltiplo de 3: lo será si lo es la suma de sus dígitos.

n	Número formado	¿Divisible?
2	12	SI

I. ¿Es múltiplo de 3?

Recordatorio del enunciado

Dado un n, construir el número formado por la concatenación de los dígitos de los números $1 \ldots n$, y decir si el resultado es múltiplo de 3: lo será si lo es la suma de sus dígitos.

n	Número formado	¿Divisible?
2	12	SI
6	123.456	SI

I. ¿Es múltiplo de 3?

Recordatorio del enunciado

Dado un n, construir el número formado por la concatenación de los dígitos de los números $1 \ldots n$, y decir si el resultado es múltiplo de 3: lo será si lo es la suma de sus dígitos.

n	Número formado	¿Divisible?
2	12	SI
6	123.456	SI
13	12.345.678.910.111.213	NO

I. ¿Es múltiplo de 3?

$sD(i)$: suma de los dígitos de i (en base 10)

I. ¿Es múltiplo de 3?

$sD(i)$: suma de los dígitos de i (en base 10)

($sD(1)$

I. ¿Es múltiplo de 3?

$sD(i)$: suma de los dígitos de i (en base 10)

($sD(1) + sD(2)$

I. ¿Es múltiplo de 3?

$sD(i)$: suma de los dígitos de i (en base 10)

$(\ sD(1) + sD(2) + sD(3))$

I. ¿Es múltiplo de 3?

$sD(i)$: suma de los dígitos de i (en base 10)

($sD(1) + sD(2) + sD(3) + \ldots + sD(13)$) % 3

I. ¿Es múltiplo de 3?

$sD(i)$: suma de los dígitos de i (en base 10)

$(\ sD(1) + sD(2) + sD(3) + \ldots + sD(13)\ )\ \%\ 3 = 55\ \%\ 3$

I. ¿Es múltiplo de 3?

$sD(i)$: suma de los dígitos de i (en base 10)

$(\ sD(1) + sD(2) + sD(3) + \ldots + sD(13)\ )\ \%\ 3 = 55\ \%\ 3 = 1$

I. ¿Es múltiplo de 3?

$sD(i)$: suma de los dígitos de i (en base 10)

$(sD(1) + sD(2) + sD(3) + \ldots + sD(13))\,\%\, 3 = 55\,\%\, 3 = 1$

Con n=13, la respuesta es `NO`

I. ¿Es múltiplo de 3?

Recordatorio del enunciado

Dado un n, construir el número formado por la concatenación de los dígitos de los números $1 \ldots n$, y decir si el resultado es múltiplo de 3 (si lo son la suma de sus dígitos)

n	Número formado	¿Divisible?
2	12	SI
6	123.456	SI
13	12.345.678.910.111.213	NO

I. ¿Es múltiplo de 3?

Recordatorio del enunciado

Dado un n, construir el número formado por la concatenación de los dígitos de los números $1 \ldots n$, y decir si el resultado es múltiplo de 3 (si lo son la suma de sus dígitos)

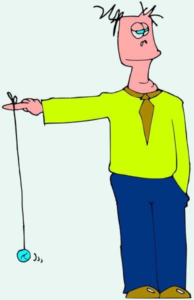
n	Número formado	¿Divisible?
2	12	SI
6	123.456	SI
13	12.345.678.910.111.213	NO
10^9	12..4.593.921.593 dígitos después..0	

I. ¿Es múltiplo de 3?

Recordatorio del enunciado

Dado un n, construir el número formado por la concatenación de los dígitos de los números $1 \ldots n$, y decir si el resultado es múltiplo de 3 (si lo son la suma de sus dígitos)

n	Número formado	¿Divisible?
2	12	SI
6	123.456	SI
13	12.345.678.910.111.213	NO
10^9	12..4.593.921.593 dígitos después..0	



I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$
1	1

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$
1	1
2	2

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$
1	1
2	2
3	3

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$
1	1
2	2
3	3
...	...
9	9

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$
1	1
2	2
3	3
...	...
9	9
10	1

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$
1	1
2	2
3	3
...	...
9	9
10	1
11	2

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$
1	1
2	2
3	3
...	...
9	9
10	1
11	2
...	...

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$
1	1
2	2
3	3
...	...
9	9
10	1
11	2
...	...
19	10

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$
1	1
2	2
3	3
...	...
9	9
10	1
11	2
...	...
19	10
20	2

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$
1	1
2	2
3	3
...	...
9	9
10	1
11	2
...	...
19	10
20	2
...	...

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

$$(sD(1) + sD(2) + sD(3) + \ldots + sD(13)) \,\%\, 3$$

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

$$(\ sD(1) + sD(2) + sD(3) + \ldots + sD(13) \ ) \,\%\, 3$$

$$(\ sD(1) \,\%\, 3 + sD(2) \,\%\, 3 + sD(3) \,\%\, 3 + \ldots + sD(13) \,\%\, 3 \ ) \,\%\, 3$$

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$	$sD(n)\,\%3$
1	1	1
2	2	2

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$	$sD(n)\,\%3$
1	1	1
2	2	2
3	3	0

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$	$sD(n)\,\%3$
1	1	1
2	2	2
3	3	0
...	...	
8	8	2
9	9	0

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$	$sD(n)\,\%3$
1	1	1
2	2	2
3	3	0
...	...	
8	8	2
9	9	0
10	1	**1**

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$	$sD(n)\,\%3$
1	1	1
2	2	2
3	3	0
...	...	
8	8	2
9	9	0
10	1	**1**
11	2	2
...	...	
19	10	1

I. ¿Es múltiplo de 3?

¿Cómo evoluciona la suma de dígitos?

n	$sD(n)$	$sD(n)\,\%3$
1	1	1
2	2	2
3	3	0
...	...	
8	8	2
9	9	0
10	1	**1**
11	2	2
...	...	
19	10	1
20	2	**2**
...	...	

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

$$(sD(1) + sD(2) + sD(3) + \ldots + sD(13))\ \%\ 3$$

$$(sD(1)\ \%\ 3 + sD(2)\ \%\ 3 + sD(3)\ \%\ 3 + \ldots + sD(13)\ \%\ 3)\ \%\ 3$$

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

$$(sD(1) + sD(2) + sD(3) + \ldots + sD(13))\ \%\ 3$$

$$(sD(1)\ \%\ 3 + sD(2)\ \%\ 3 + sD(3)\ \%\ 3 + \ldots + sD(13)\ \%\ 3)\ \%\ 3$$

¡Eureka!

$$(1 + 2 + 0 + 1 + 2 + 0 + \ldots + sD(n)\ \%\ 3)\ \%\ 3$$

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

$$(sD(1) + sD(2) + sD(3) + \ldots + sD(13)) \% 3$$

$$(sD(1) \% 3 + sD(2) \% 3 + sD(3) \% 3 + \ldots + sD(13) \% 3) \% 3$$

¡Eureka!

$$(1 + 2 + 0 + 1 + 2 + 0 + \ldots + sD(n) \% 3) \% 3$$

Podemos hacer más módulos. . .

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

$$(sD(1) + sD(2) + sD(3) + \ldots + sD(13)) \% 3$$

$$(sD(1) \% 3 + sD(2) \% 3 + sD(3) \% 3 + \ldots + sD(13) \% 3) \% 3$$

¡Eureka!

$$(1 + 2 + 0 + 1 + 2 + 0 + \ldots + sD(n) \% 3) \% 3$$

Podemos hacer más módulos. . .

¡Requete-eureka!

$$¡ (1 + 2) \% 3 = 0 !$$

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

¡Lo importante es cómo termina la serie de sumas!

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

¡Lo importante es cómo termina la serie de sumas!

- $(1 + 2 + ... + 1 + 2 + \mathbf{0})\,\%\;3$

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

¡Lo importante es cómo termina la serie de sumas!

- $(1 + 2 + \ldots + 1 + 2 + \mathbf{0})\,\%\,3 = 0$

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

¡Lo importante es cómo termina la serie de sumas!

- $(1 + 2 + \ldots + 1 + 2 + \mathbf{0})\,\%\,3 = 0 \rightarrow$ Divisible por 3

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

¡Lo importante es cómo termina la serie de sumas!

- $(1 + 2 + \ldots + 1 + 2 + \mathbf{0})\ \%\ 3 = 0 \rightarrow$ Divisible por 3
- $(1 + 2 + \ldots + 1 + 2 + 0 + \mathbf{1})\ \%\ 3$

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

¡Lo importante es cómo termina la serie de sumas!

- $(1 + 2 + \ldots + 1 + 2 + \mathbf{0})\,\%\,3 = 0 \rightarrow$ Divisible por 3
- $(1 + 2 + \ldots + 1 + 2 + 0 + \mathbf{1})\,\%\,3 = 1 \rightarrow$ No es divisible por 3

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

¡Lo importante es cómo termina la serie de sumas!

- $(1 + 2 + \ldots + 1 + 2 + \mathbf{0})\ \%\ 3 = 0 \rightarrow$ Divisible por 3
- $(1 + 2 + \ldots + 1 + 2 + 0 + \mathbf{1})\ \%\ 3 = 1 \rightarrow$ No es divisible por 3
- $(1 + 2 + \ldots + 1 + 2 + 0 + 1 + \mathbf{2})\ \%\ 3$

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

¡Lo importante es cómo termina la serie de sumas!

- $(1 + 2 + \ldots + 1 + 2 + \mathbf{0})\ \%\ 3 = 0 \rightarrow$ Divisible por 3
- $(1 + 2 + \ldots + 1 + 2 + 0 + \mathbf{1})\ \%\ 3 = 1 \rightarrow$ No es divisible por 3
- $(1 + 2 + \ldots + 1 + 2 + 0 + 1 + \mathbf{2})\ \%\ 3 = 0 \rightarrow$ Divisible por 3

I. ¿Es múltiplo de 3?

Aritmética modular al rescate

¡Lo importante es cómo termina la serie de sumas!

- $(1 + 2 + \ldots + 1 + 2 + \mathbf{0})\,\%\,3 = 0 \rightarrow$ Divisible por 3
- $(1 + 2 + \ldots + 1 + 2 + 0 + \mathbf{1})\,\%\,3 = 1 \rightarrow$ No es divisible por 3
- $(1 + 2 + \ldots + 1 + 2 + 0 + 1 + \mathbf{2})\,\%\,3 = 0 \rightarrow$ Divisible por 3

Entonces...

. . . sí; basta con calcular el módulo de n y 3 ☺

D. Amigo invisible

Envíos	Válidos	% éxito
32	6	18 %

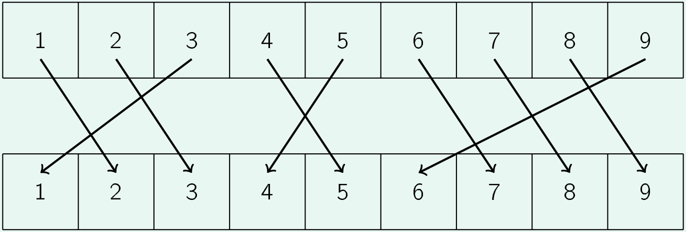
D. Amigo invisible

Recordatorio del enunciado

Dada las asignaciones conocidas de un *amigo invisible*, ¿podemos averiguar el resto?

D. Amigo invisible

La asignación es una permutación.



Se pueden analizar los *ciclos* y pensar en permutaciones.

D. Amigo invisible

Como un grafo

- Una persona es un nodo.
- Una arista une el que regala con el regalado.
- En la asignación completa, cada nodo tiene una arista que sale, y una arista que entra.
- No puede haber autoaristas (nadie se regala a sí mismo).
- Si hay n nodos, tiene que haber n aristas.

D. Amigo invisible

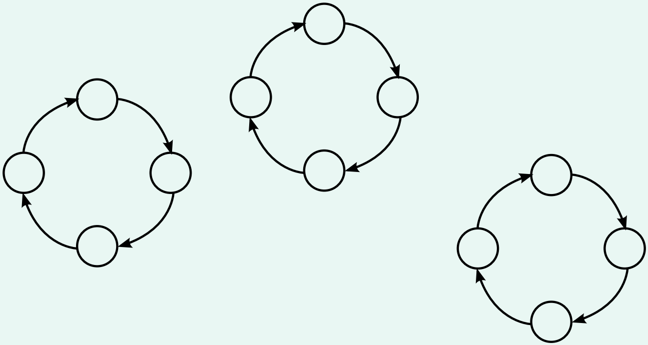
Como un grafo

- Una persona es un nodo.
- Una arista une el que regala con el regalado.
- En la asignación completa, cada nodo tiene una arista que sale, y una arista que entra.
- No puede haber autoaristas (nadie se regala a sí mismo).
- Si hay n nodos, tiene que haber n aristas.

Analizamos las opciones dependiendo de cuántas aristas falten

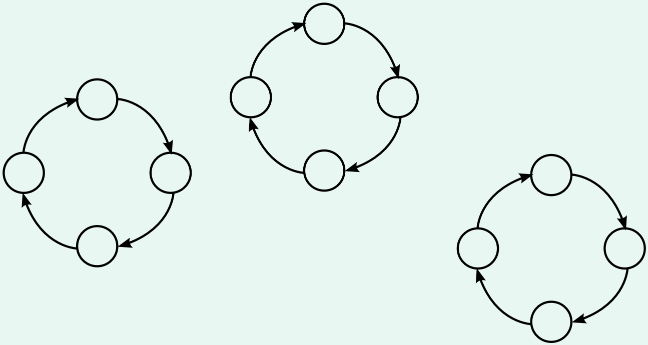
D. Amigo invisible

Faltan 0 aristas



D. Amigo invisible

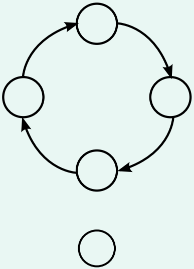
Faltan 0 aristas



Lo sabemos todo

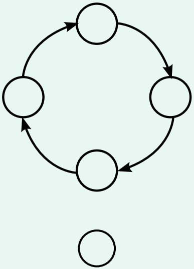
D. Amigo invisible

Falta 1 aristas



D. Amigo invisible

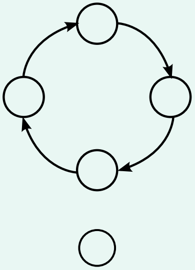
Falta 1 aristas



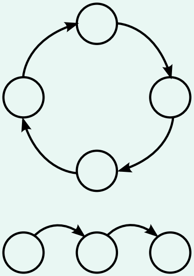
Caso inválido

D. Amigo invisible

Falta 1 aristas

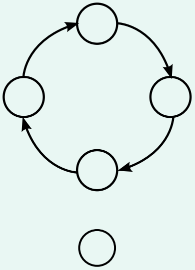


Caso inválido

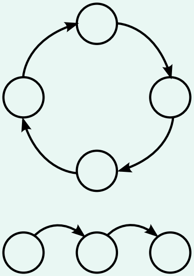


D. Amigo invisible

Falta 1 aristas



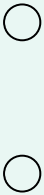
Caso inválido



Asignación única

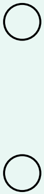
D. Amigo invisible

Faltan 2 aristas



D. Amigo invisible

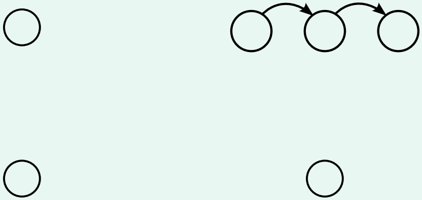
Faltan 2 aristas



Asignación única

D. Amigo invisible

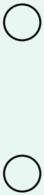
Faltan 2 aristas



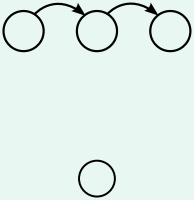
Asignación única

D. Amigo invisible

Faltan 2 aristas



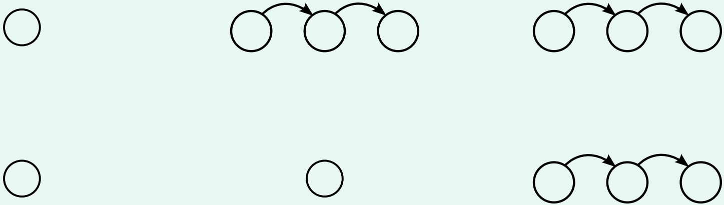
Asignación única



Asignación única

D. Amigo invisible

Faltan 2 aristas

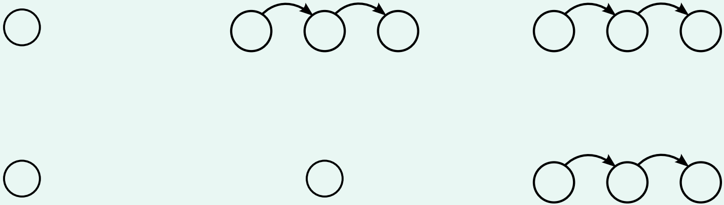


Asignación única

Asignación única

D. Amigo invisible

Faltan 2 aristas



Asignación única   Asignación única   Asignación múltiple

D. Amigo invisible

Faltan 3 aristas

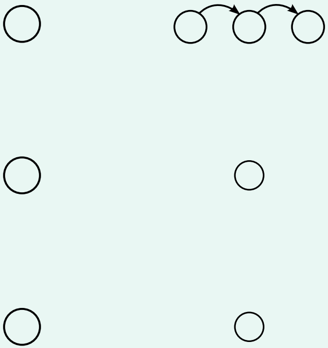
D. Amigo invisible

Faltan 3 aristas

Múltiple

D. Amigo invisible

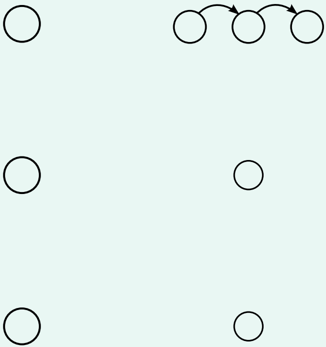
Faltan 3 aristas



Múltiple

D. Amigo invisible

Faltan 3 aristas

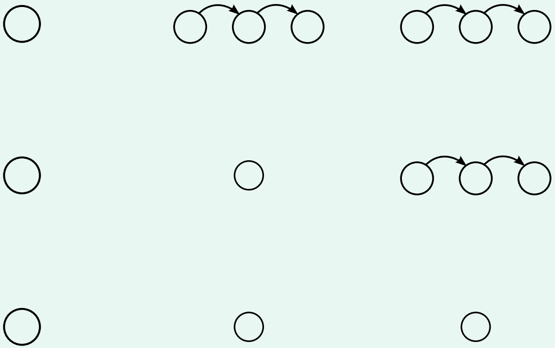


Múltiple

Múltiple

D. Amigo invisible

Faltan 3 aristas

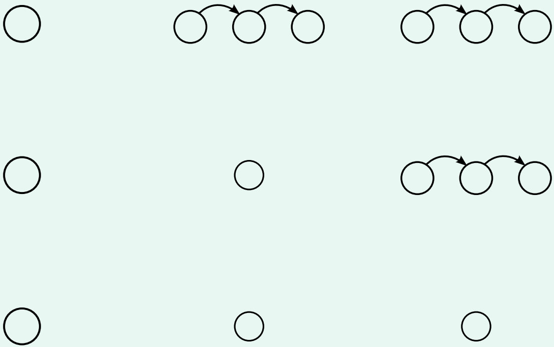


Múltiple

Múltiple

D. Amigo invisible

Faltan 3 aristas



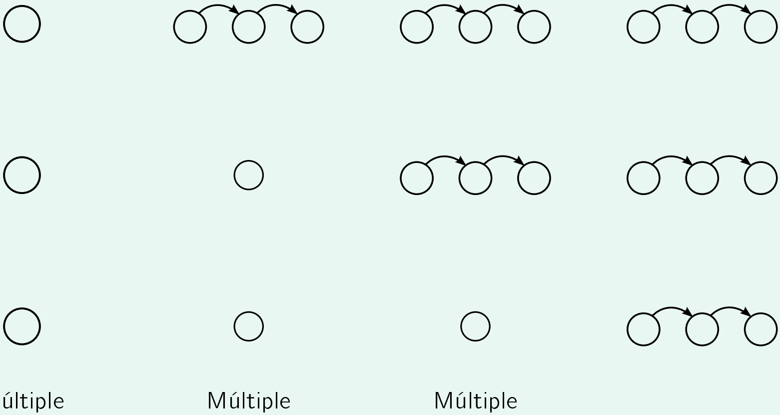
Múltiple

Múltiple

Múltiple

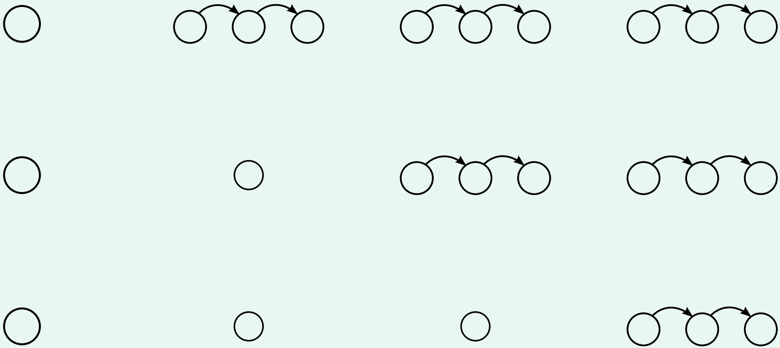
D. Amigo invisible

Faltan 3 aristas



D. Amigo invisible

Faltan 3 aristas



Múltiple Múltiple Múltiple Múltiple

D. Amigo invisible

Solución

Controlar los diferentes casos según los grados de entrada y salida, y contestar lo que corresponda.

```
    ada
+ byron
-------
  molar
```

A. Aritmética verbal

Envíos	Válidos	% éxito
2	0	0 %

A. Aritmética verbal

Recordatorio del enunciado

$$\begin{array}{rr} & \texttt{ada} \\ + & \texttt{byron} \\ \hline & \texttt{molar} \end{array} \Rightarrow \begin{array}{rr} & 767 \\ + & 39508 \\ \hline & 40275 \end{array}$$

A. Aritmética verbal

Primera idea

```
      ada
+   byron
---------
    molar
```

A. Aritmética verbal

Primera idea

```
     ada
+  byron
--------
   molar
```

letra	dígito
a	0
b	1
d	2
l	3
m	4
n	5
o	6
r	7
y	8

A. Aritmética verbal

Primera idea

```
   ada
+  byron
--------
   molar
```

letra	dígito
a	0
b	1
d	2
l	3
m	4
n	5
o	6
r	7
y	8

```
   020
+  18765
--------
   46307
```

A. Aritmética verbal

Primera idea

$$\begin{array}{r} \texttt{ada} \\ +\ \texttt{byron} \\ \hline \texttt{molar} \end{array}$$

letra	dígito
a	0
b	1
d	2
l	3
m	4
n	5
o	6
r	7
y	9

$$\begin{array}{r} 020 \\ +\ 19765 \\ \hline 46307 \end{array}$$

A. Aritmética verbal

Primera idea

```
  ada
+ byron
-------
 molar
```

letra	dígito
a	0
b	1
d	2
l	3
m	4
n	5
o	6
r	8
y	9

```
   020
+ 19865
-------
  46308
```

A. Aritmética verbal

Primera idea

Un rato después...

A. Aritmética verbal

Primera idea

$$\begin{array}{r} \texttt{ada} \\ + \ \texttt{byron} \\ \hline \texttt{molar} \end{array}$$

letra	dígito
a	7
b	3
d	6
l	2
m	4
n	8
o	0
r	5
y	9

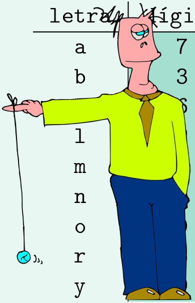
$$\begin{array}{r} 767 \\ + \ 39508 \\ \hline 40275 \end{array}$$

A. Aritmética verbal

Primera idea

```
    ada
+ byron
-------
  molar
```

letra	dígito
a	7
b	3
	6
l	
m	
n	
o	
r	
y	



```
    767
+ 39508
-------
  40275
```

A. Aritmética verbal

Pensando un poco...

```
    ada
+ byron
-------
  molar
```

letra	dígito
a	0
b	1
d	2
l	3
m	4
n	5
o	6
r	7
y	8

```
    020
+ 18765
-------
  46307
```

A. Aritmética verbal

Pensando un poco...

$$
\begin{array}{r}
\texttt{ada} \\
+\ \texttt{byron} \\
\hline
\texttt{molar}
\end{array}
$$

letra	dígito
a	0
b	1
d	2
l	3
m	4
n	5
o	6
r	7
y	8

$$
\begin{array}{r}
020 \\
+\ 18765 \\
\hline
46307
\end{array}
$$

0+5 nunca será 7 por mucho que toquemos los dígitos de la izquierda

A. Aritmética verbal

Pensando un poco...

```
    ada
+ byron
-------
  molar
```

letra	dígito
a	
b	
d	
l	
m	
n	
o	
r	
y	

A. Aritmética verbal

Pensando un poco...

```
    0d0
+ byro1
-------
  mola2
```

letra	dígito
a	0
b	
d	
l	
m	
n	1
o	
r	2
y	

A. Aritmética verbal

Pensando un poco...

Un instante después...

A. Aritmética verbal

Pensando un poco...

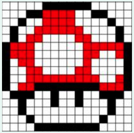
```
      1
    7d7
+ byro8
-------
  mola5
```

letra	dígito
a	7
b	
d	
l	
m	
n	8
o	
r	5
y	

A. Aritmética verbal

Vuelta atrás

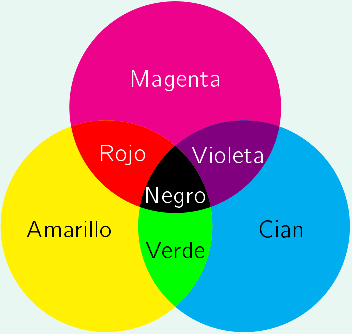
- Asignar dígitos a letras *por columnas*, *volviendo atrás* si se llega a un resultado parcial incorrecto.
  - No puede haber ceros *superfluos* a la izquierda.
    - Cuidado con el caso $0+0=0$
  - Al asignar una letra a un dígito, puede que se fijen números más adelante.
  - En la última “columna” no puede haber acarreo.
- Es más fácil de programar si se da la vuelta a palabras (se está siempre en el mismo índice).



B. Pixel Art

Envíos	Válidos	% éxito
54	35	64 %

B. Pixel Art



1. Decrementar los contadores de colores primarios disponibles hasta que uno de ellos se haga negativo (caso `NO`) o bien se termine la entrada (caso `SI`).
2. Calcular la cantidad de colores primarios necesarios para pintar la entrada; después comprobar si hay suficiente de cada uno de ellos y cuánto sobra en ese caso.

F. Hijos a tope

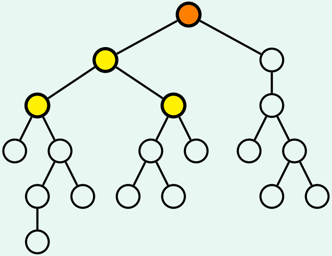
Envíos	Válidos	% éxito
11	7	63 %

F. Hijos a tope

- Generalizar la función y calcular dos resultados:
  1. *total*: el número de generaciones de la subfamilia generacionalmente completa con mayor número de generaciones.
  2. *raiz*: el número de generaciones de la subfamilia generacionalmente completa con mayor número de generaciones **que cuelga de la raíz**.

$$raiz \leq total$$

$raiz = 1$ $total = 2$

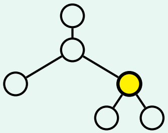


F. Hijos a tope

- Caso base: un nodo $raiz = 0$ $total = 0$

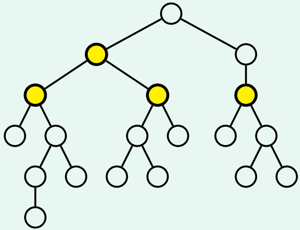
- Caso recursivo: con un hijo

$$raiz = 0 \qquad total = 1 \rightarrow \text{ el del hijo}$$



- Caso recursivo: con dos hijos

$$raiz = min\{raizI, raizD\} + 1 \qquad total = max\{totalI, totalD, raiz\}$$



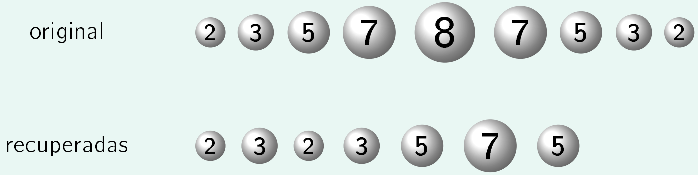
K. Las perlas de la condesa

Envíos	Válidos	% éxito
60	24	40 %

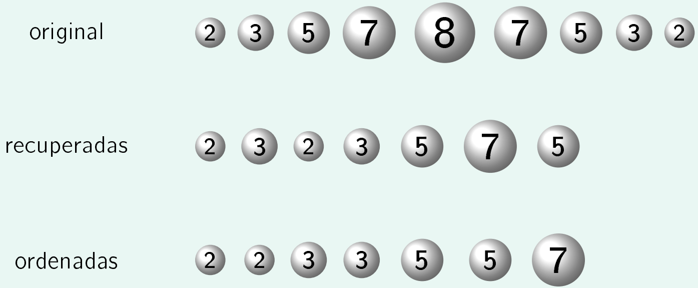
K. Las perlas de la condesa

original

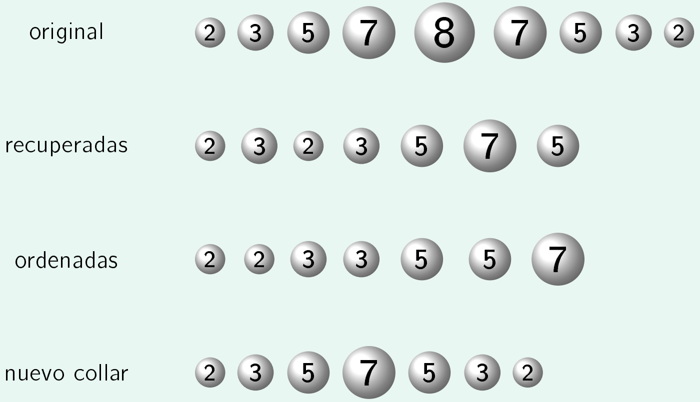
K. Las perlas de la condesa



K. Las perlas de la condesa



K. Las perlas de la condesa



G. Chocolate con almendras

Envíos	Válidos	% éxito
2	1	50 %

G. Chocolate con almendras

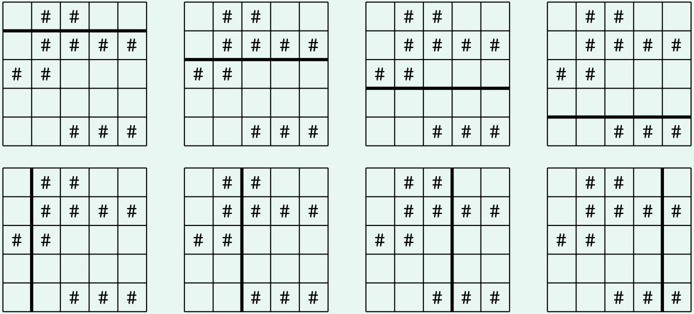
	#	#		
	#	#	#	#
#	#			
		#	#	#

Encontrar el mínimo número de cortes que hacen falta para separar las onzas con almendras de las que no las tienen.

G. Chocolate con almendras

	#	#		
	#	#	#	#
#	#			
		#	#	#

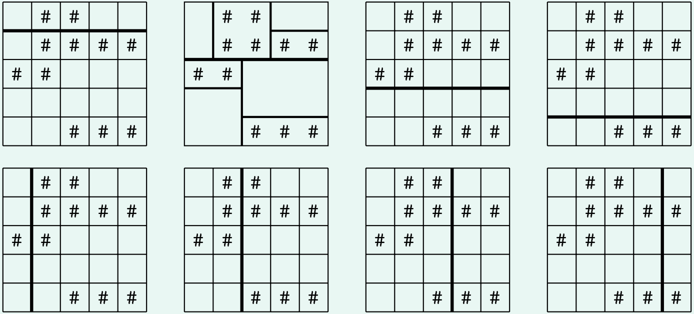
Encontrar el mínimo número de cortes que hacen falta para separar las onzas con almendras de las que no las tienen.



G. Chocolate con almendras

	#	#		
	#	#	#	#
#	#			
		#	#	#

Encontrar el mínimo número de cortes que hacen falta para separar las onzas con almendras de las que no las tienen.



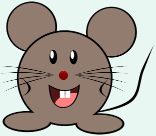
G. Chocolate con almendras

$$
\begin{aligned}
cortes(f_1,f_2,c_1,c_2) &= 0 \qquad \text{si solo hay un sabor}\\
cortes(f_1,f_2,c_1,c_2) &= \min\Big(\min_{f_1 \le i < f_2}\{cortes(f_1,i,c_1,c_2) + cortes(i+1,f_2,c_1,c_2)\},\\
&\qquad \min_{c_1 \le j < c_2}\{cortes(f_1,f_2,c_1,j) + cortes(f_1,f_2,j+1,c_2)\}\Big) + 1
\end{aligned}
$$

G. Chocolate con almendras

$$
\begin{aligned}
cortes(f_1, f_2, c_1, c_2) &= 0 \qquad \text{si solo hay un sabor} \\
cortes(f_1, f_2, c_1, c_2) &= \min\Big(\min_{f_1 \leq i < f_2} \{cortes(f_1, i, c_1, c_2) + cortes(i+1, f_2, c_1, c_2)\}, \\
&\qquad \min_{c_1 \leq j < c_2} \{cortes(f_1, f_2, c_1, j) + cortes(f_1, f_2, j+1, c_2)\}\Big) + 1
\end{aligned}
$$

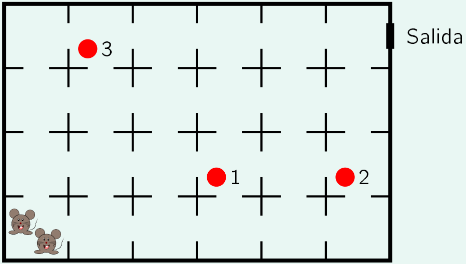
Implementar con programación dinámica (*top-down*).



- J. Más listos que el hambre

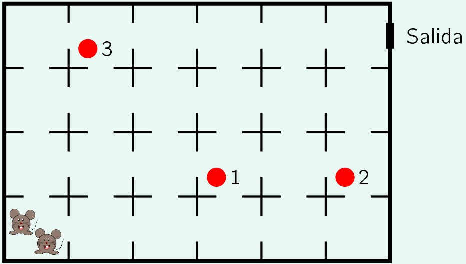
Envíos	Válidos	% éxito
7	0	0 %

J. Más listos que el hambre



Encontrar la forma en la que los dos ratones tienen que colaborar para pulsar los botones en orden en el menor tiempo posible.

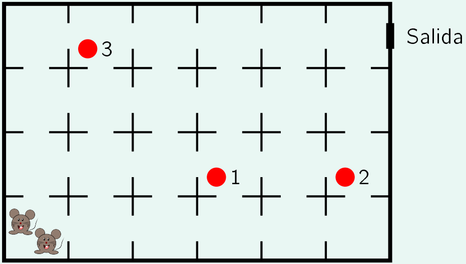
J. Más listos que el hambre



Encontrar la forma en la que los dos ratones tienen que colaborar para pulsar los botones en orden en el menor tiempo posible.

Posibles estados: r_1, r_2

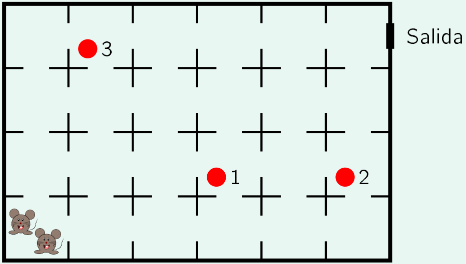
J. Más listos que el hambre



Encontrar la forma en la que los dos ratones tienen que colaborar para pulsar los botones en orden en el menor tiempo posible.

Posibles estados: r_1, r_2, p

J. Más listos que el hambre

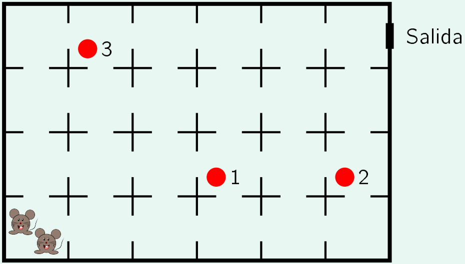


Encontrar la forma en la que los dos ratones tienen que colaborar para pulsar los botones en orden en el menor tiempo posible.

Posibles estados: r_1, r_2, p

Siguientes estados: se mueve el ratón en r_1: $r_1 + 1, r_2, p + dist(r_1, r_1 + 1)$

J. Más listos que el hambre



Encontrar la forma en la que los dos ratones tienen que colaborar para pulsar los botones en orden en el menor tiempo posible.

Posibles estados: r_1, r_2, p

Siguientes estados: se mueve el ratón en r_1: $r_1 + 1, r_2, p + dist(r_1, r_1 + 1)$
o se mueve el ratón en r_2: $r_1 + 1, r_1, \max(0, dist(r_2, r_1 + 1) - p)$

J. Más listos que el hambre

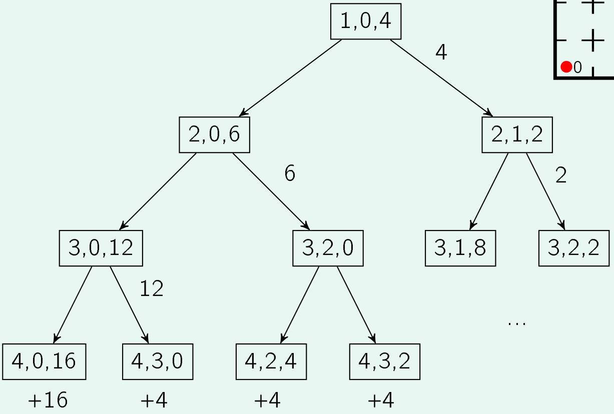
Programación dinámica

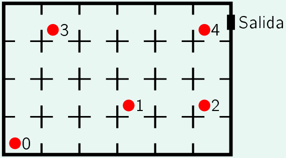
$$
\begin{array}{rcl}
ratones(r_1, r_2, p) & = & \max(p, dist(r_2, salida)) \quad \text{si } r_1 \text{ es la salida} \\
ratones(r_1, r_2, p) & = & \min(ratones(r_1+1, r_2, p + dist(r_1, r_1+1)), \\
& & \qquad ratones(r_1+1, r_1, p') + p) \quad \text{e.o.c.} \\
& & \text{donde } p' = \max(0, dist(r_2, r_1+1) - p)
\end{array}
$$

Llamada inicial: $ratones(1, 0, dist(comienzo, 1))$

J. Más listos que el hambre

Camino mínimo en el grafo (implícito) de estados

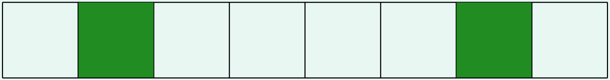




L. Durmiendo en albergues

Envíos	Válidos	% éxito
68	23	33 %

L. Durmiendo en albergues



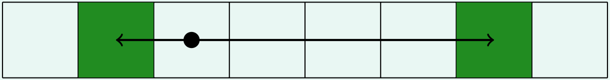
Dadas las ocupaciones de camas en un albergue con las camas en hileras, ¿en qué cama duermes para mantenerte lo más alejado posible de cualquier otro huésped?

L. Durmiendo en albergues

Solución 1: Para cada posición se puede calcular cuál es el vecino más cercano, recorriendo el vector hacia la izquierda y derecha.

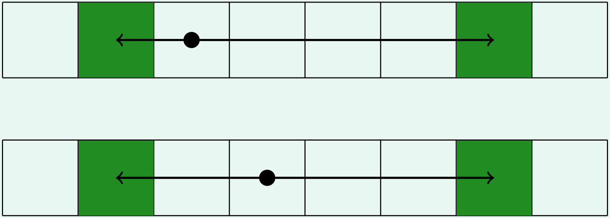
L. Durmiendo en albergues

Solución 1: Para cada posición se puede calcular cuál es el vecino más cercano, recorriendo el vector hacia la izquierda y derecha.



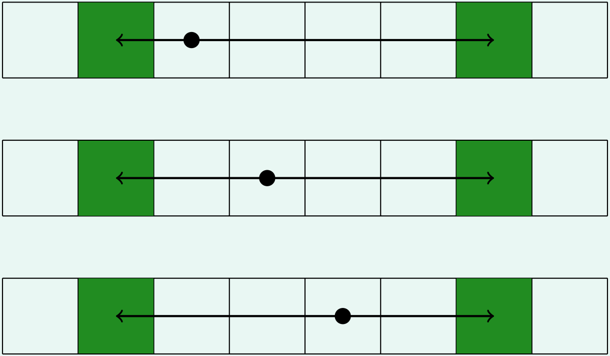
L. Durmiendo en albergues

Solución 1: Para cada posición se puede calcular cuál es el vecino más cercano, recorriendo el vector hacia la izquierda y derecha.



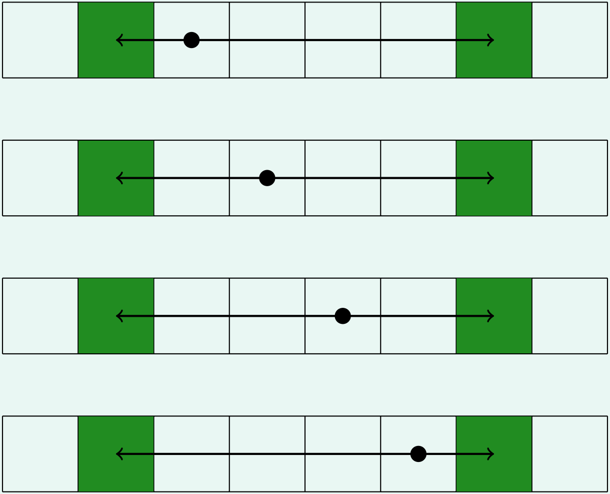
L. Durmiendo en albergues

Solución 1: Para cada posición se puede calcular cuál es el vecino más cercano, recorriendo el vector hacia la izquierda y derecha.



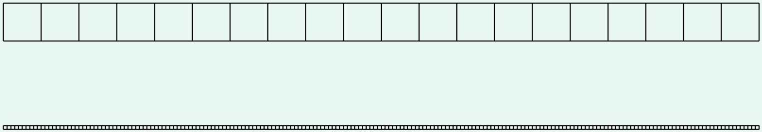
L. Durmiendo en albergues

Solución 1: Para cada posición se puede calcular cuál es el vecino más cercano, recorriendo el vector hacia la izquierda y derecha.



L. Durmiendo en albergues

L. Durmiendo en albergues



L. Durmiendo en albergues

¡Lento!

Por cada posición puedes necesitar recorrer el albergue/vector entero. Con hasta 500.000 camas, eso significa TLE.

L. Durmiendo en albergues

Solución 2: recorrer el vector midiendo las longitudes de los intervalos sin camas, quedándose con la más grande, l.

L. Durmiendo en albergues

Solución 2: recorrer el vector midiendo las longitudes de los intervalos sin camas, quedándose con la más grande, l.

La solución es $l/2$.

L. Durmiendo en albergues

Solución 2: recorrer el vector midiendo las longitudes de los intervalos sin camas, quedándose con la más grande, l.

La solución es $l/2$.

Cuidado: en los intervalos de los extremos no necesitamos ponermos en el centro.

Se tratan de forma especial.

L. Durmiendo en albergues

Otras soluciones:

- En Java `line.split("X")`.

- H. Las partituras de la orquesta

Envíos	Válidos	% éxito
56	5	8 %

H. Las partituras de la orquesta

En una orquesta, distribuir los p atriles entre los distintos instrumentos para minimizar el número de músicos agolpados en cada instrumento.

H. Las partituras de la orquesta

Solución 1 - Estrategia voraz

```
Dar un atril/partitura a cada instrumento
Mientras queden atriles/partituras por repartir
    Dar un atril al instrumento con más músicos por partitura

Escribir la respuesta
```

H. Las partituras de la orquesta

Solución 1 - Estrategia voraz

```
Dar un atril/partitura a cada instrumento
Mientras queden atriles/partituras por repartir
    Dar un atril al instrumento con más músicos por partitura

Escribir la respuesta
```

Uso de una cola de prioridad de máximos con una entrada por instrumento ordenada por la relación musicos/atriles.

H. Las partituras de la orquesta

Solución 2 - Búsqueda binaria en el espacio de soluciones

Si admitimos un máximo de K personas en un atril, ¿tenemos atriles para todos?

H. Las partituras de la orquesta

Solución 2 - Búsqueda binaria en el espacio de soluciones

Si admitimos un máximo de K personas en un atril, ¿tenemos atriles para todos?

- Si $K = 0$, no.
- Si $K = n$, sí.
- Entre 0 y n está la respuesta.

H. Las partituras de la orquesta

Solución 2 - Búsqueda binaria en el espacio de soluciones

Si admitimos un máximo de K personas en un atril, ¿tenemos atriles para todos?

- Si $K = 0$, no.
- Si $K = n$, sí.
- Entre 0 y n está la respuesta.

Solución

Hacemos búsqueda binaria de la respuesta

H. Las partituras de la orquesta

Fallos

- En la solución voraz, no se puede dividir el dato leído entre dos y volverlo a meter

```
11 4
4 6 7 8
```

- E. El entretenimiento de las máquinas

Envíos	Válidos	% éxito
-	-	-

E. El entretenimiento de las máquinas

Contar el número de apariciones de una colección de palabras en una sopa de letras.

E. El entretenimiento de las máquinas

Contar el número de apariciones de una colección de palabras en una sopa de letras.

Posible solución

```
Para cada posición en la sopa
   Para cada dirección
      Para cada palabra
         si está la palabra
             apariciones[palabra]++
```

E. El entretenimiento de las máquinas

Entrada

- Tamaño de la sopa: hasta 200 x 200
- Número de palabras: hasta 20.000
- Tamaño de cada palabra: hasta 200 (independientemente del tamaño de la sopa)
- **Ocho** direcciones

E. El entretenimiento de las máquinas

Entrada

- Tamaño de la sopa: hasta 200 x 200
- Número de palabras: hasta 20.000
- Tamaño de cada palabra: hasta 200 (independientemente del tamaño de la sopa)
- **Ocho** direcciones

¡Lento!

La solución anterior es demasiado lenta.

E. El entretenimiento de las máquinas

- Se debe utilizar alguna estructura que reduzca la complejidad.

E. El entretenimiento de las máquinas

- Se debe utilizar alguna estructura que reduzca la complejidad.
- Hasta 20.000 palabras en el diccionario. Preferible eliminar ese factor.

E. El entretenimiento de las máquinas

- Se debe utilizar alguna estructura que reduzca la complejidad.
- Hasta 20.000 palabras en el diccionario. Preferible eliminar ese factor.
- Almacenar en alguna estructura todas las palabras

E. El entretenimiento de las máquinas

Almacenar en una estructura todo el diccionario de forma que para cada posición se busquen *a la vez* todas las palabras.

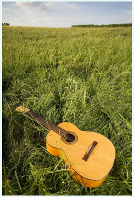
Solución

Estructura de árbol binario: **trie**

E. El entretenimiento de las máquinas

Alternativas

- Se puede añadir al diccionario/trie las palabras invertidas, para ahorrar los recorridos en cuatro de las direcciones.



- C. El profesor de música

Envíos	Válidos	% éxito
–	–	–

C. El profesor de música

Enunciado: Encontrar el pueblo donde de muda el profesor de música para minimizar los kilómetros recorridos.

Enunciado alternativo: Dado un grafo no dirigido, encontrar el nodo que haga que el ciclo que pasa por él y otros k nodos dados tenga longitud mínima.

C. El profesor de música

La respuesta depende de:

- El orden en el que se visitan los k nodos.
- El nodo seleccionado.

C. El profesor de música

La respuesta depende de:

- El orden en el que se visitan los k nodos.
- El nodo seleccionado.

```
para cada camino que visite los k nodos
   para cada nodo/pueblo
      calcular la longitud del ciclo
```

C. El profesor de música

Consideraciones

- Los nodos fijados son como mucho 6
  - $6! = 720$ caminos distintos posibles (o menos si eliminamos simetrías)
- Los nodos distintos son hasta 5.000.

C. El profesor de música

Consideraciones

- Los nodos fijados son como mucho 6
  - $6! = 720$ caminos distintos posibles (o menos si eliminamos simetrías)
- Los nodos distintos son hasta 5.000.
- Si la comprobación para cada posibilidad es rápida, es factible probar todas las posibilidades (hasta $720 * 4994$).

C. El profesor de música

Solución

- Para cada pueblo con colegio, precalcular la distancia mínima con el resto de pueblos
  - Hasta 6 ejecuciones de Dijkstra por consulta
- Para cada par [camino, nodo], utilizar esa información precalculada para sacar la longitud del camino.
- Quedarse con la más corta.

C. El profesor de música

Solución mejorada

- Para cada pueblo con colegio, precalcular la distancia mínima con el resto de pueblos
    - Hasta 6 ejecuciones de Dijkstra por consulta
- Para cada posible [inicio, fin] de camino, calcular la longitud del camino más corto.
- Para cada [inicio, fin, nodo], utilizar esa información precalculada para calcular la longitud del camino.
- Quedarse con la más corta.

En lugar de $6!$ comprobaciones por cada nodo, habrá $6 * 5$.