

X AdaByron - Madrid 2024

Estadísticas y Soluciones

inetum
Positive digital flow

next digital

rtve
Cátedra RTVE-UC3M

VALCU

uc3m | Universidad Carlos III de Madrid
Departamento de Informática

Clasificación de los problemas

Problema	Categoría
A - Final de Peón vs Rey	Adhoc, Pensar.
B - Huelga de Agricultores	Grafos, Flujo, MinCut
C - Y sin embargo se mueve	Queries, LCA
D - El problema de las 3 fuentes	Geometría
E - La cola del supermercado	Programación Dinámica
F - Follow the LIDAR	Interactivo
G - El pensador cuadriculado	Ad-hoc
H - Mini Tetris	Programación Dinámica
I - El calcetín mágico	Teoría de Juegos
J - Walcu: Explotando globos	Estructuras de Datos, Colas de prioridad
K - La conjetura de Go*d*dbach	Aritmética
L - Next Digital: Espacios de Coworking	Grafos, Floyd

Estadísticas

Problema	# casos de prueba	Espacio en disco
A - Final de Peón vs Rey	20936	586 KB
B - Huelga de Agricultores	47	217 KB
C - Y sin embargo se mueve	89	81,8 MB
D - El problema de las 3 fuentes	10016	512 KB
E - La cola del supermercado	407	119 KB
F - Follow the LIDAR	8	2,69 KB
G - El pensador cuadriculado	200003	2,69 MB
H - Mini Tetris	22	11,2 MB
I - El calcetín mágico	462045	185 MB
J - Walcu: Explotando globos	2014	12,3 MB
K - La conjetura de Go*d*dbach	500003	7,92 MB
L - Next Digital: Espacios de Coworking	107	1,19 MB
- **Total**	1.195.697	303,53 MB (+-)

Estadísticas*

Problema	Primer equipo en resolverlo	Tiempo
A - Final de Peón vs Rey	The Fan Team	93
B - Huelga de Agricultores	Max Fraud	121
C - Y sin embargo se mueve	Vexed Hamsters .. Logic	202
D - El problema de las 3 fuentes	-	-
E - La cola del supermercado	$\pi k2's$	16
F - Follow the LIDAR	Braquistócronos	50
G - El pensador cuadriculado	char izard;	115
H - Mini Tetris	Literally Nefarious	110
I - El calcetín mágico	-	-
J - Walcu: Explotando globos	Los Churreros	38
K - La conjetura de Go*d*dbach	ScanfSupremacy	38
L - Next Digital: Espacios de Coworking	Max Fraud	33

*Antes de congelar el marcador.

Dificultad de los problemas según los jueces

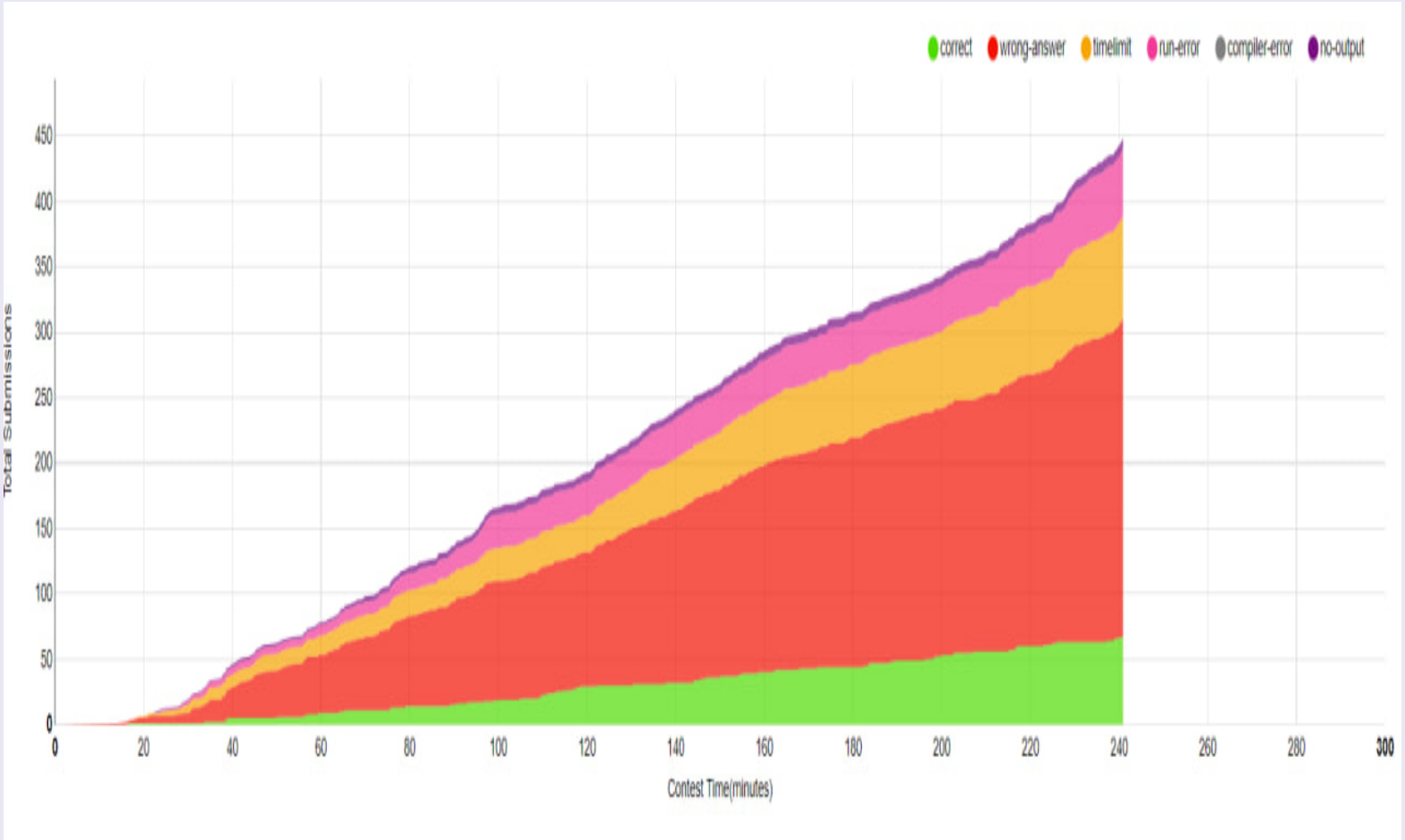
Problema	Categoría
A - Final de Peón vs Rey	Baja (2)
B - Huelga de Agricultores	Alta (9)
C - Y sin embargo se mueve	Muy alta (11)
D - El problema de las 3 fuentes	Alta (10)
E - La cola del supermercado	Media (5)
F - Follow the LIDAR	Media (8)
G - El pensador cuadriculado	Baja (1)
H - Mini Tetris	Media (7)
I - El calcetín mágico	Muy alta (12)
J - Walcu: Explotando globos	Media (6)
K - La conjetura de Go*d*dbach	Media (3)
L - Next Digital: Espacios de Coworking	Media (4)

Estadísticas*

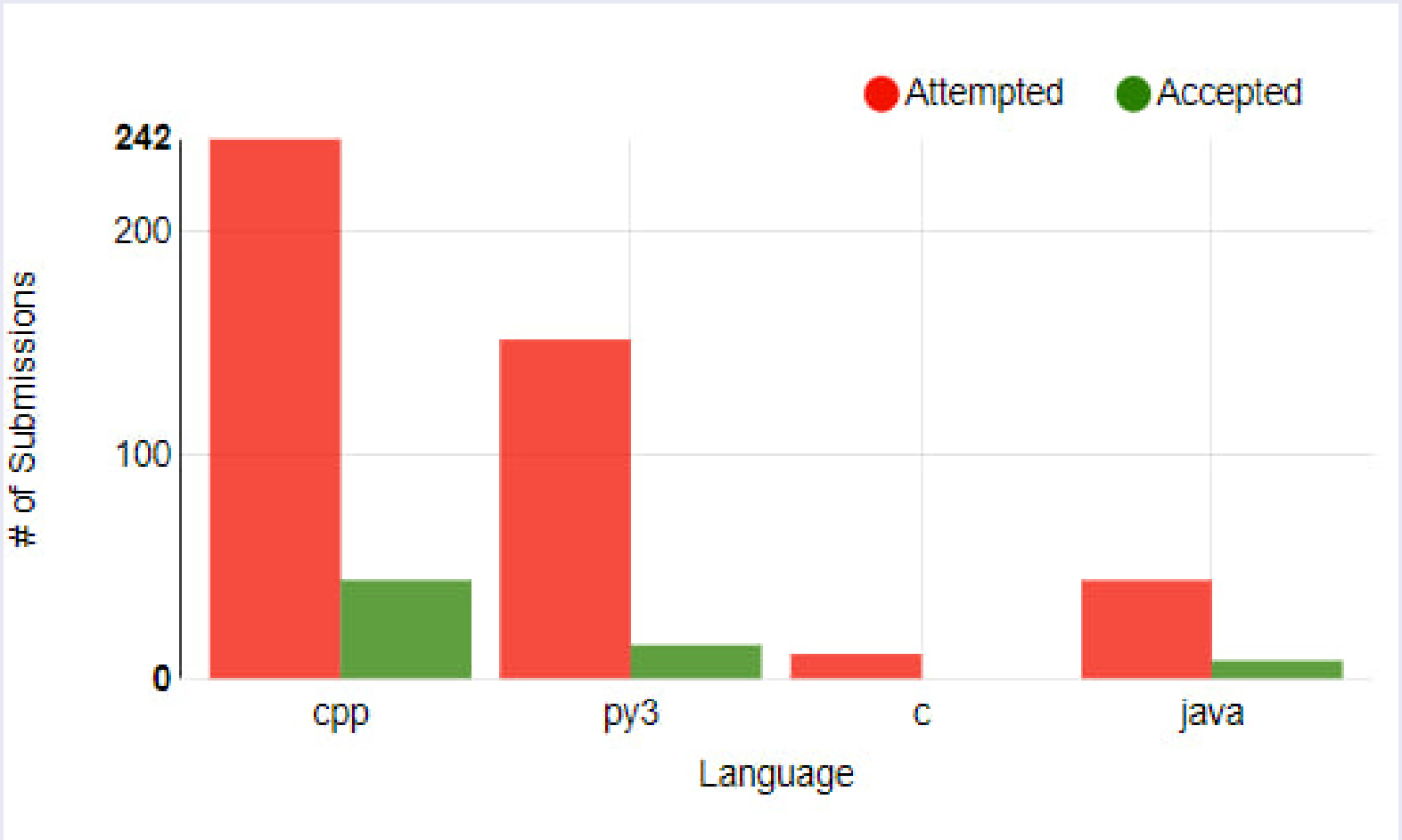
Problema	Envíos	Válidos	% éxito
A - Final de Peón vs Rey	74	6	8 %
B - Huelga de Agricultores	5	2	40 %
C - Y sin embargo se mueve	10	3	30 %
D - El problema de las 3 fuentes	8	0	0 %
E - La cola del supermercado	48	7	14 %
F - Follow the LIDAR	18	2	11 %
G - El pensador cuadriculado	27	11	40 %
H - Mini Tetris	6	3	50 %
I - El calcetín mágico	8	0	0 %
J - Walcu: Explotando globos	157	10	6 %
K - La conjetura de Go*d*dbach	43	15	34 %
L - Next Digital: Espacios de Coworking	38	7	15 %

*Antes de congelar el marcador.

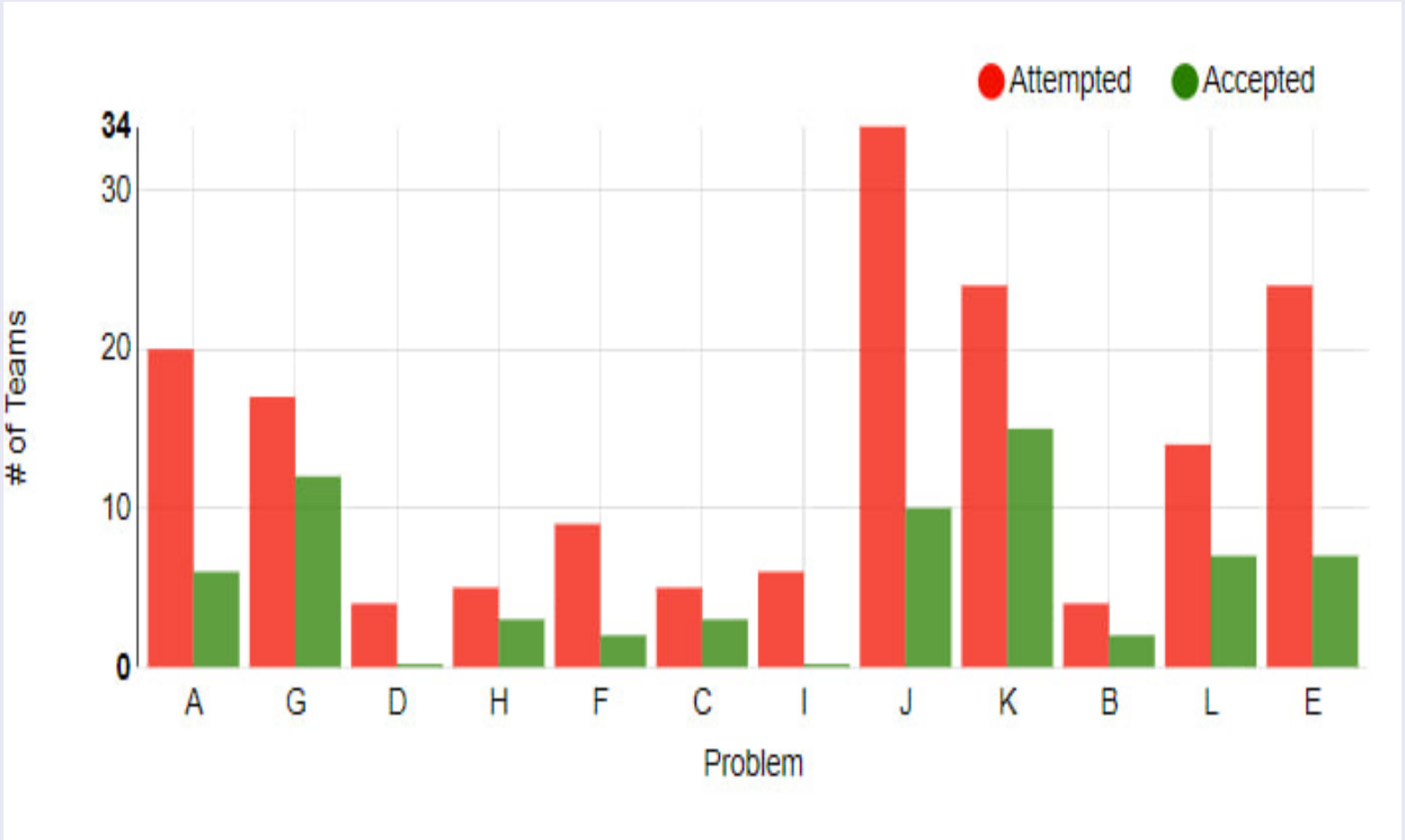
Estadísticas varias



Estadísticas varias



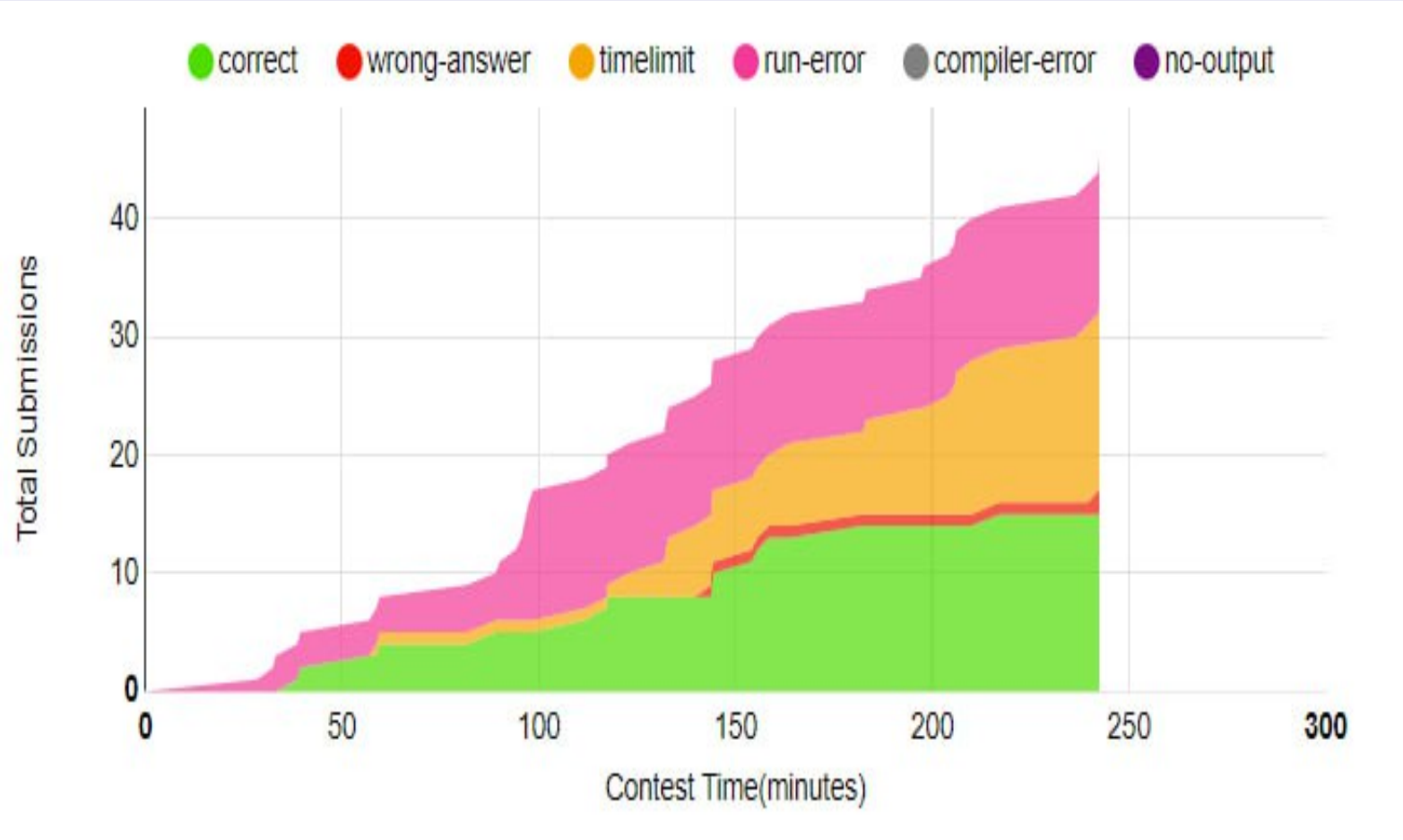
Estadísticas varias



K. La conjetura de Go*d*dbach

Envíos	Válidos	% éxito
43	15	34.88 %

K. La conjetura de Go*d*dbach



K. La conjetura de Go*d*dbach

"*Cualquier número impar mayor que 1, que no sea primo, puede describirse como la suma de dos números primos*"
Si un número impar resulta de la suma de dos números, uno debe ser necesariamente par, y el único que es primo, es el 2.
Por lo tanto, un número n pertenece a la serie de Go*d*dbach $\mathcal{G}$ si y sólo si $n-2$ es un número primo.

K. La conjetura de Go*d*dbach

Dada una serie de números primos $\mathcal{P}$, calculada con la *criba de Eratóstenes*:

```
prime[i]=true, ∀i
for i ← 2 to B do
    if prime[i] then
        P ← i
        for j ← 2i to B do
            prime[j]=false
            j += i
        end
    end
end
```

O, con números tan pequeños, usando Rabin-Miller.

K. La conjetura de Go*d*dbach

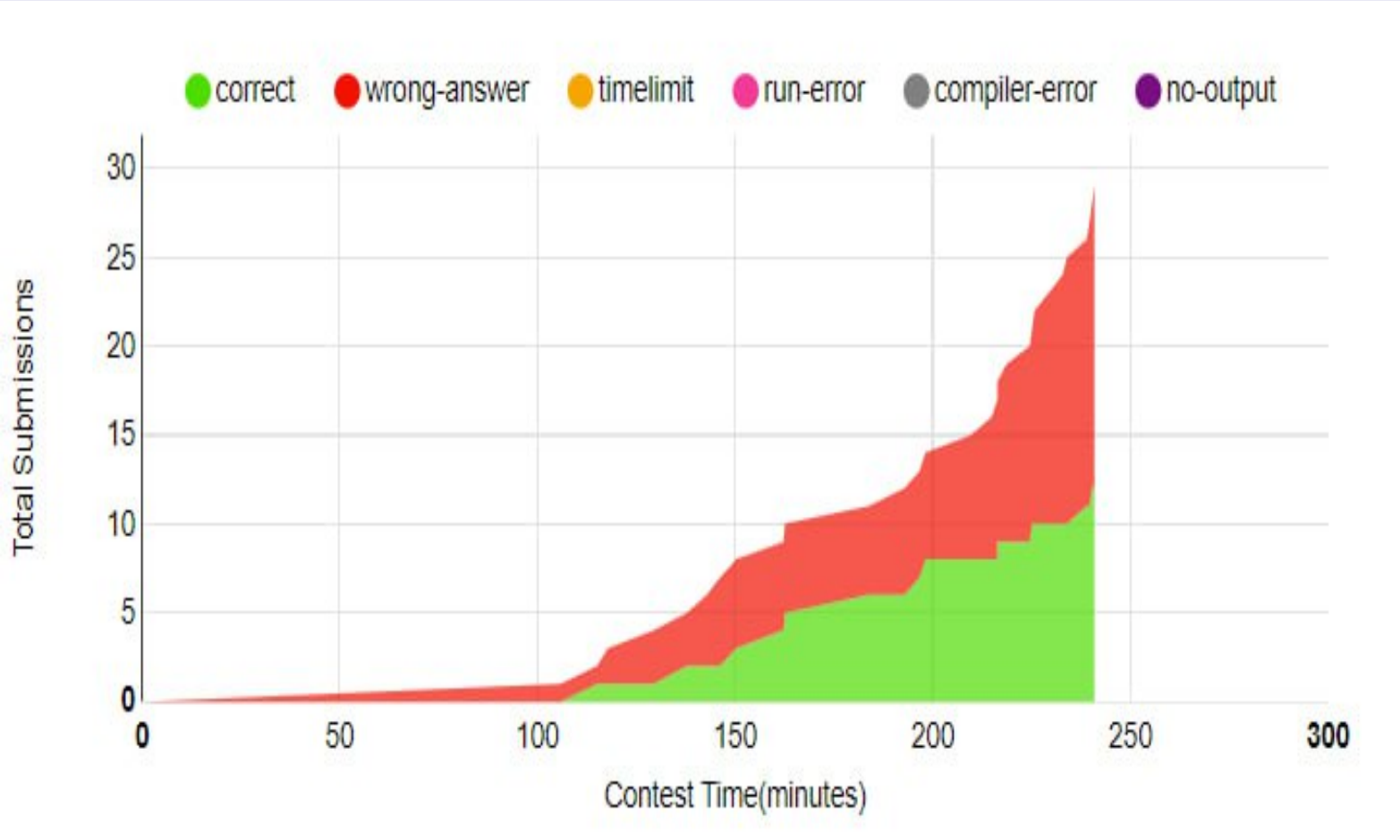
Basta con recorrerla contando posiciones si $n-2$ es primo ($\mathcal{P}_i = n-2$), y n no lo sea ($\mathcal{P}_{i+1} > n$), hasta devolver el de la posición indicada.

```
items ← 0
for i ← 0 to |P| do
    if 2 + P_i < P_{i+1} then
        items++
    end
    if items == index then
        return 2 + P_i
    end
end
```

G. El pensador cuadriculado

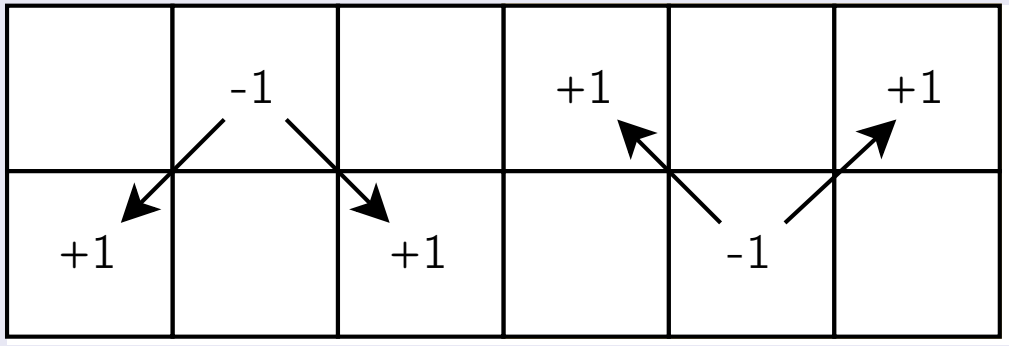
Envíos	Válidos	% éxito
27	11	40.74 %

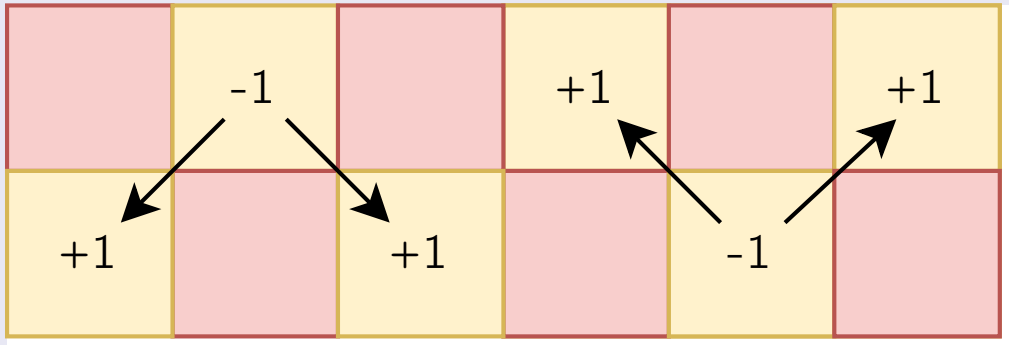
G. El pensador cuadriculado



G. El pensador cuadriculado

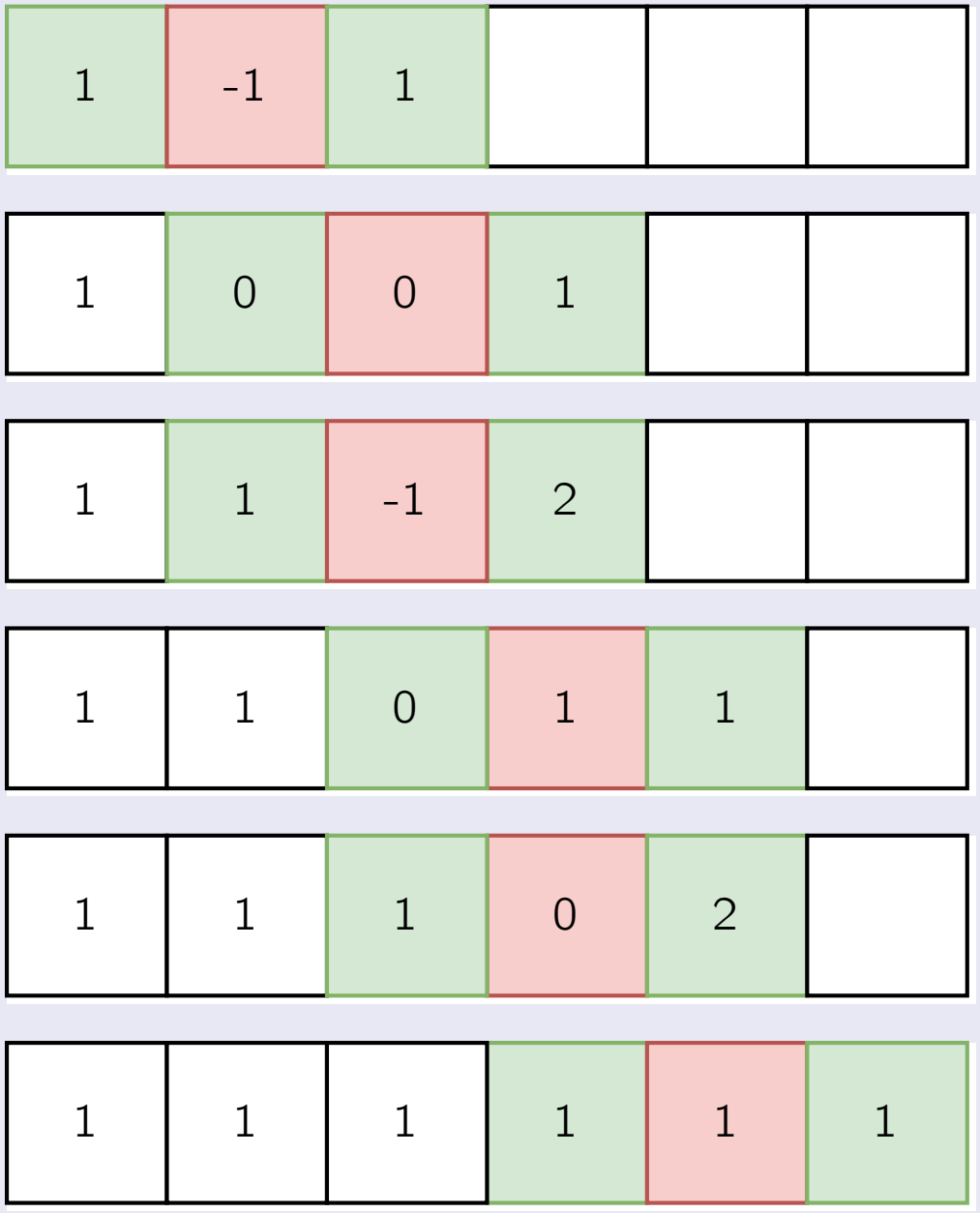
Operaciones en una cuadrícula $2 \times M$. En una operación, se coge una casilla, se le resta 1, y se le suma 1 a sus dos vecinas diagonales. Solo se puede si las dos están en la cuadrícula.





Se transforma en dos subproblemas de cuadrículas $1 \times M$.

G. El pensador cuadriculado



G. El pensador cuadriculado

- El orden de las operaciones da igual, son sumas (propiedad conmutativa).
- Cada 6 operaciones "forzadas" por un extremo, se ponen 6 unos, uno en cada casilla, y no antes.
- Si queremos más casillas llenas o con mayor valor, el problema se repite, cada 6 operaciones.
- Si M no es múltiplo de 6, el proceso anterior no puede acabar correctamente a la vez para todas las casillas.

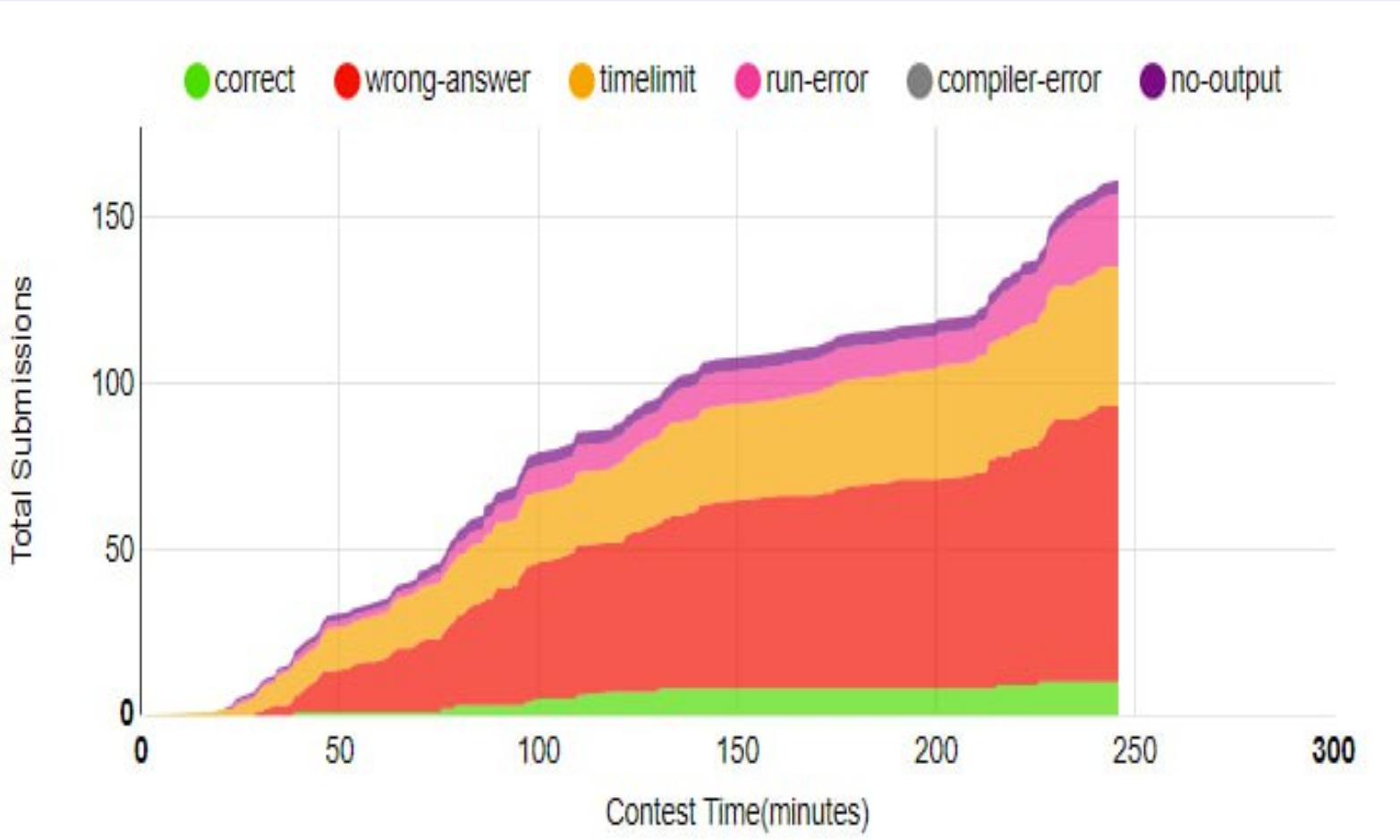
G. El pensador cuadriculado

```
for _ in range(int(input)):
    M, V = map(int, input().split())
    if M % 6 != 0:
        print(-1)
    else:
        print(2 * M * V)
```

J. Walcu: Explotando globos

Envíos	Válidos	% éxito
157	10	6.36 %

J. Walcu: Explotando globos



J. Walcu: Explotando globos

¿Cuánto suman los A números más pequeños y los B números más grandes de un conjunto al que se van añadiendo números?

J. Walcu: Explotando globos

Caso 1 ($A = 1, B = 1$):

1 2 3

J. Walcu: Explotando globos

Caso 1 ($A = 1, B = 1$):

1 2 3 4

J. Walcu: Explotando globos

Caso 1 ($A = 1, B = 1$):

1 2 3 4 5

J. Walcu: Explotando globos

Lo que **NO** funcionaba / trampas del problema:

- Mantener todo en un array ordenado -> TLE
  - Ordenar solo cuando se pregunta -> TLE
- Reducir el número de valores a guardar -> WA
- No utilizar enteros de 64 bits -> WA
- Hacer el módulo al final -> WA

J. Walcu: Explotando globos

Cosas a tener en cuenta:

- Hay hasta 10^5 eventos
- Los números son muy grandes (10^{18})
- Aritmética modular

Ninguna operación (H o P) puede tener coste lineal -> TLE

J. Walcu: Explotando globos

Solución:

- Dividir los números que se tienen en 2 conjuntos A y B
  - A contiene las a menores cantidades.
  - B contiene las b mayores cantidades.
- Se calcula S, la suma de los valores contenidos en A y B, y se actualiza cada vez que entre o salga un elemento de alguno de los conjuntos.
- Cuando ocurre un evento de tipo H:
  - Si el elemento más grande de A es mayor que el nuevo valor (x), se introduce x en A y se elimina el mayor valor de A.
  - Si el elemento más pequeño de B es menor que el nuevo valor (x), se introduce x en B y se elimina el menor valor de B.
- Cuando ocurre un evento de tipo P, se imprime el valor de S.

J. Walcu: Explotando globos

¿Qué se puede utilizar para guardar los datos de los conjuntos A y B?

- Colas de prioridad
- Mapas ordenados de conteo
- Multiconjuntos...

J. Walcu: Explotando globos

¿Qué se puede utilizar para guardar los datos de los conjuntos A y B?

- Colas de prioridad
- Mapas ordenados de conteo
- Multiconjuntos...

Todo esto hace que todas las operaciones de tipo P tengan un coste en tiempo de $O(log(E))$

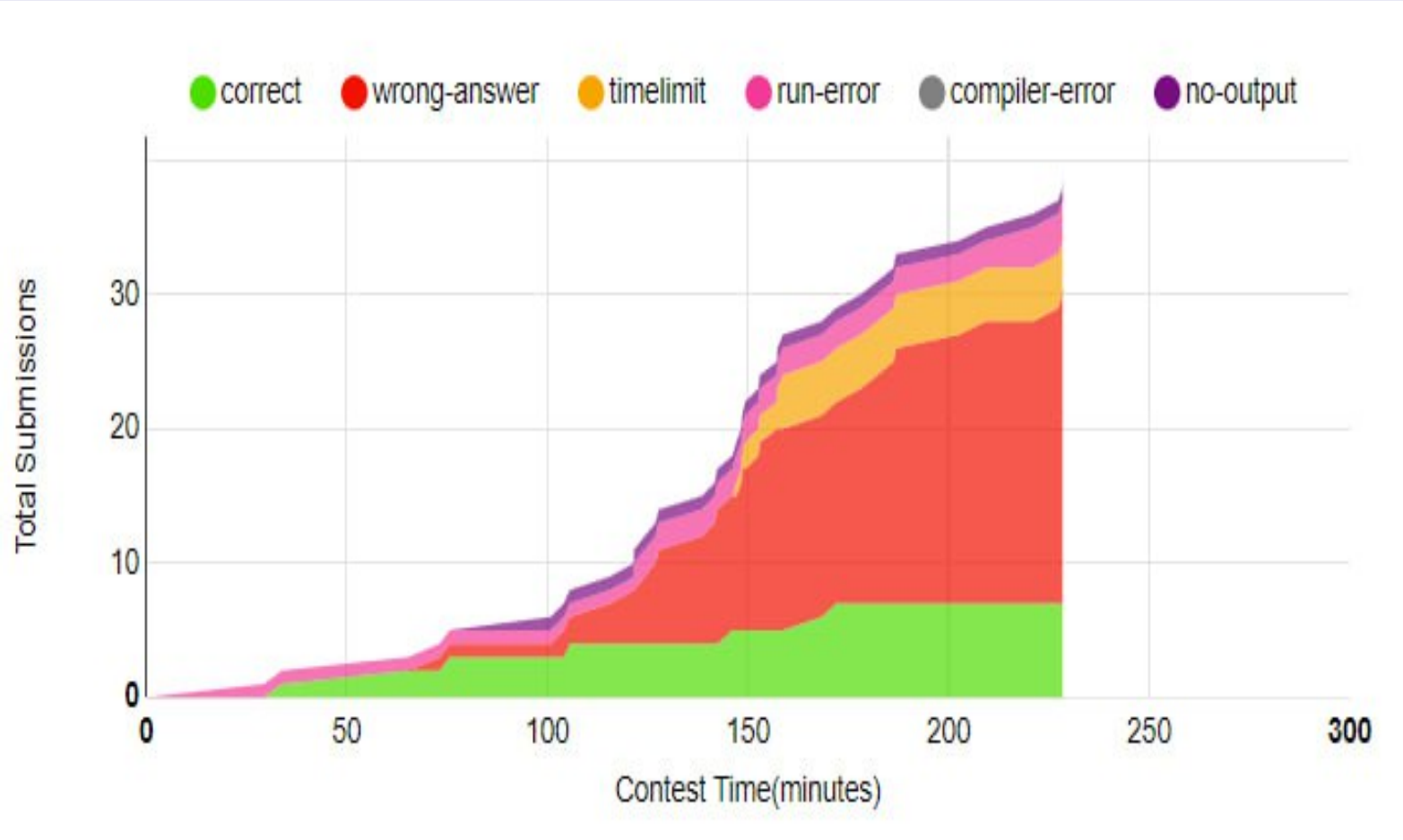
Complejidad de la solución: $O(E * log(E))$

L. NEXT DIGITAL: Espacios de Coworking

Envíos	Válidos	% éxito
38	7	18.5 %



L. NEXT DIGITAL: Espacios de Coworking



L. NEXT DIGITAL: Espacios de Coworking

Somos muchos nexters distribuidos por toda España y parte del extranjero y, aunque somos 100 % remotos, queremos ofrecer la posibilidad de tener un espacio de trabajo a todos los nexters.

Tenemos las ciudades dónde vivimos y el tiempo del medio de transporte más rápido que comunica alguna de ellas. Necesitamos validar un listado de empresas de espacios de coworking comprobando si tienen sedes suficientemente bien distribuidas de forma que ningún nexter esté T o más minutos en transporte.

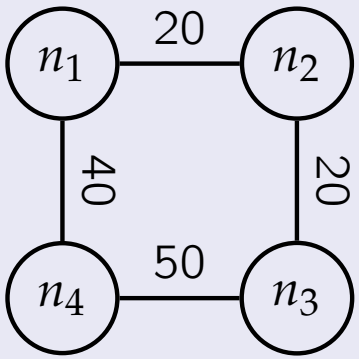
L. NEXT DIGITAL: Espacios de Coworking

Somos muchos nexters distribuidos por toda España y parte del extranjero y, aunque somos 100 % remotos, queremos ofrecer la posibilidad de tener un espacio de trabajo a todos los nexters.

Tenemos ~~las ciudades dónde vivimos~~ NODOS y ~~el tiempo del medio de transporte más rápido~~ ARISTAS que comunica alguna de ellas. Necesitamos validar un listado de empresas de espacios de coworking comprobando si tienen sedes suficientemente bien distribuidas de forma que ningún nexter esté T o más minutos en transporte.

L. NEXT DIGITAL: Espacios de Coworking

$T = 60$

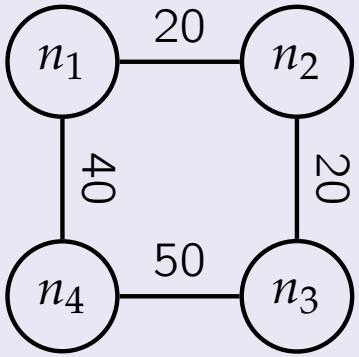


Sedes:

1. n_2
2. n_3
3. n_4
4. n_1 n_2

L. NEXT DIGITAL: Espacios de Coworking

$T = 60$



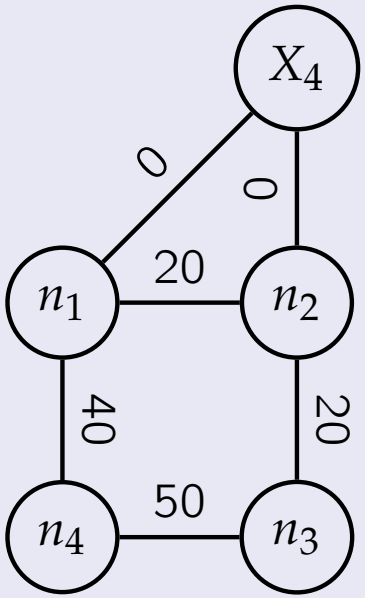
	n_1	n_2	n_3	n_4
n_1	0	20	40	40
n_2	20	0	20	60
n_3	40	20	0	50
n_4	40	60	50	0

Sedes:

1. n_2
2. n_3
3. n_4
4. n_1 n_2

L. NEXT DIGITAL: Espacios de Coworking

$T = 60$



Sedes:

1. n_2
2. n_3
3. n_4
4. n_1 n_2

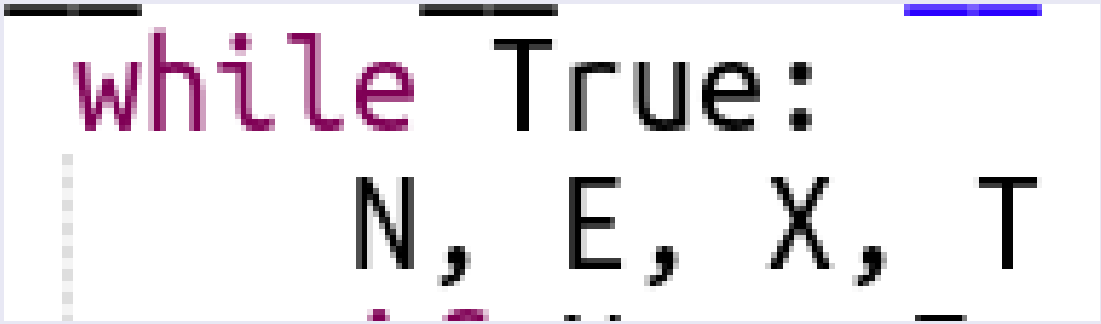
L. NEXT DIGITAL: Espacios de Coworking

PERO

- E puede ser 0.
- X puede ser 0.
- Se pide $< T$ y no $\leq T$.

L. NEXT DIGITAL: Espacios de Coworking

Por siempre <3 NEXT



- πk2's
- Max Fraud
- Pan de Pita
- ScanfSupremacy
- Vexed Hamsters Doing Logic
- 418 I'm a teapot
- ACalorados

E. La cola del supermercado

Envíos	Válidos	% éxito
48	7	14.58 %

E. La cola del supermercado

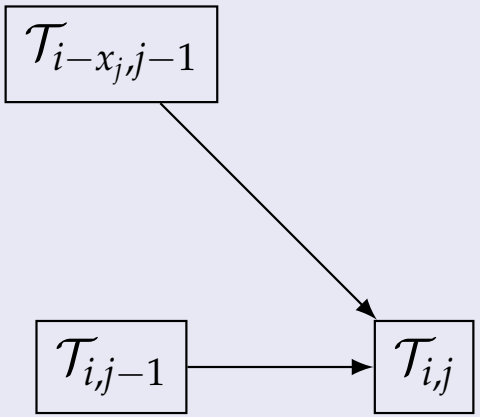
Se da un conjunto de números $\mathcal{S} = \{x_0, x_1, \ldots, x_{n-1}\}$, y el propósito consiste en dividirlos en dos subconjuntos, $S_1, S_2 \subset \mathcal{S}$, $S_1 \cap S_2 = \varnothing$, tal que la diferencia de su suma sea la mínima: $\textbf{mín} \left| \sum_{x_j \in S_1} - \sum_{x_k \in S_2} \right|$

Y este problema es un Subset Sum Problem (SSP) donde el objetivo es:

$$T = \frac{1}{2} \sum_{x_i \in \mathcal{S}}$$

E. La cola del supermercado

SSP se resuelve en tiempo pseudopolinomial con *Programación Dinámica* usando una tabla $\mathcal{T}$, donde $\mathcal{T}_{ij}$ es la suma más cercana a i usando los primeros j valores de $\mathcal{S}$:



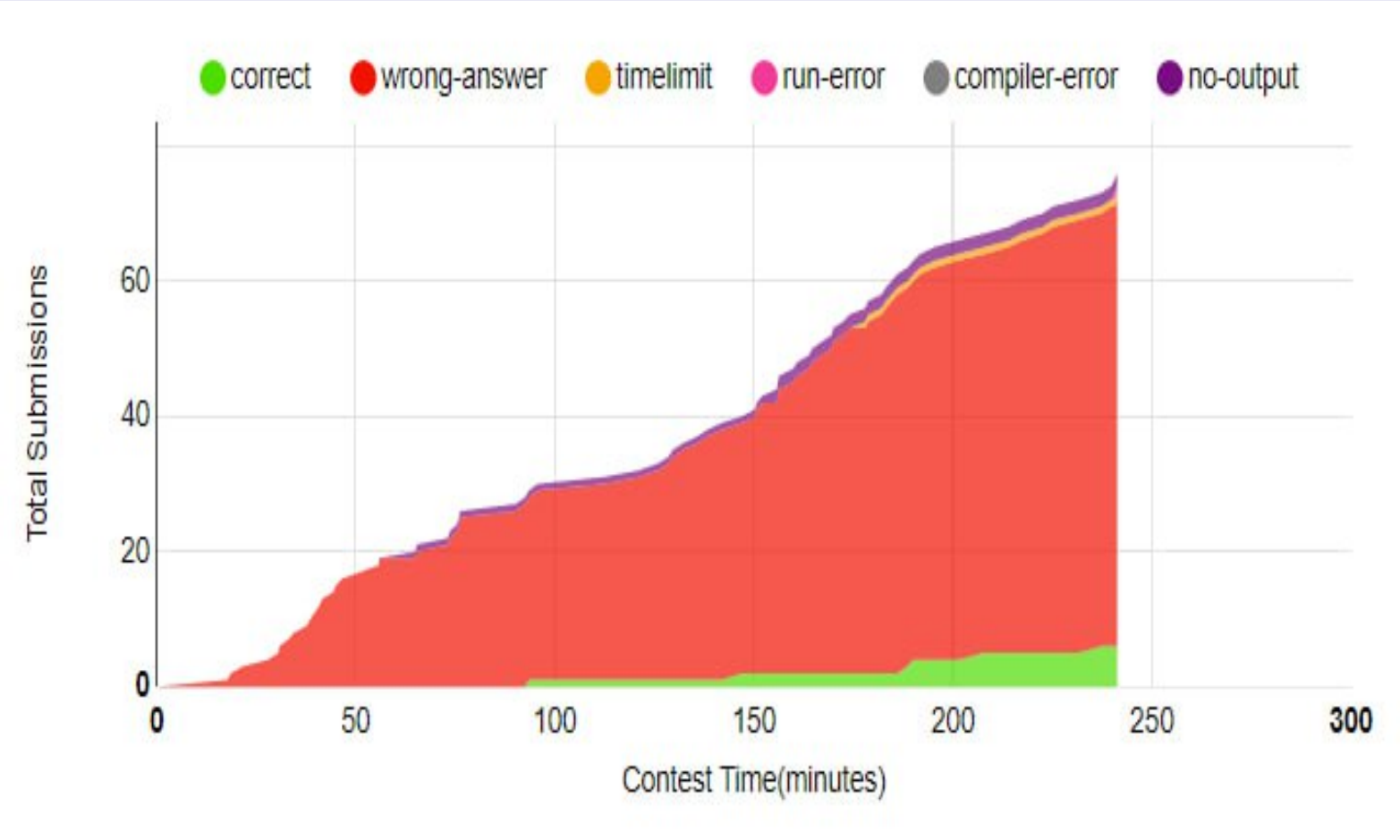
$$\mathcal{T}_{\substack{ij \\ 1\leq i\leq T, 0\leq j\leq n-1}} = \begin{cases} \mathcal{T}_{i,j-1} & \text{si } |\mathcal{T}_{i,j-1} - T| \leq |\mathcal{T}_{i-x_j,j-1} + x_j - T| \\ \mathcal{T}_{i-x_j,j-1} + x_j & \text{en otro caso} \end{cases}$$

Y la solución es $\mathcal{T}_{T,n-1}$

A. Final de Peón vs Rey

Envíos	Válidos	% éxito
74	6	8.11 %

A. Final de Peón vs Rey

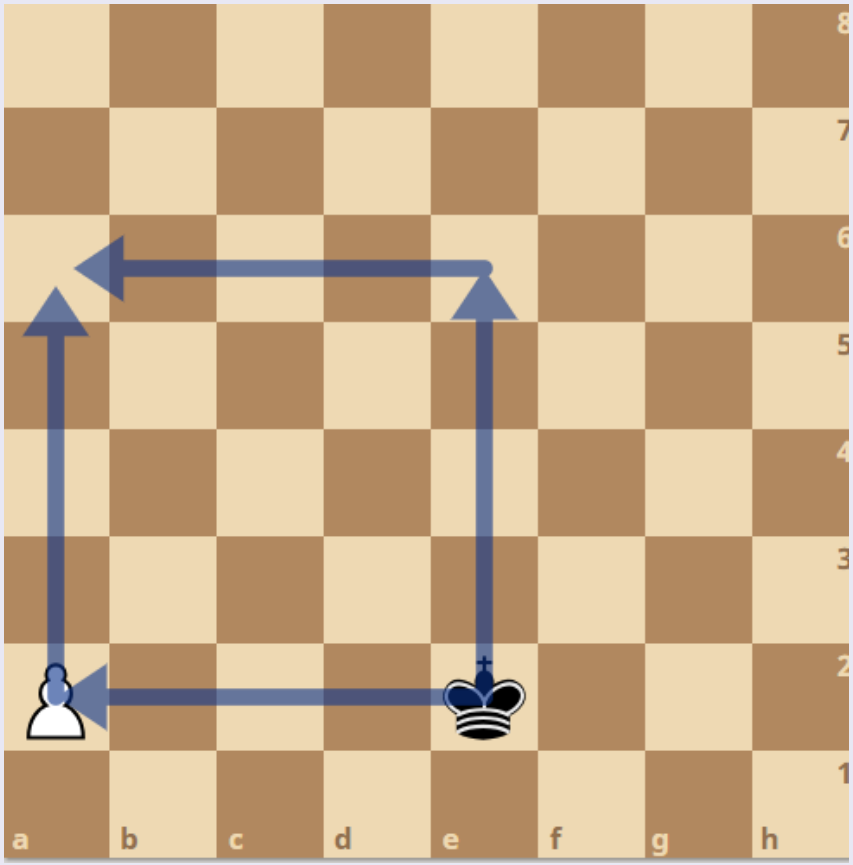


A. Final de Peón vs Rey

Dado un tablero de ajedrez de $N \cdot N$, determinar si el rey es capaz de capturar al peón.
Para simplificar el problema, se asume que solo existe un rey. No existen posiciones ilegales (el peón está bloqueado y le toca mover, el rey está en jaque y le toca al rival, etc).
Esto en ajedrez se resuelve con la regla del "cuadrado".

A. Final de Peón vs Rey

Regla general..

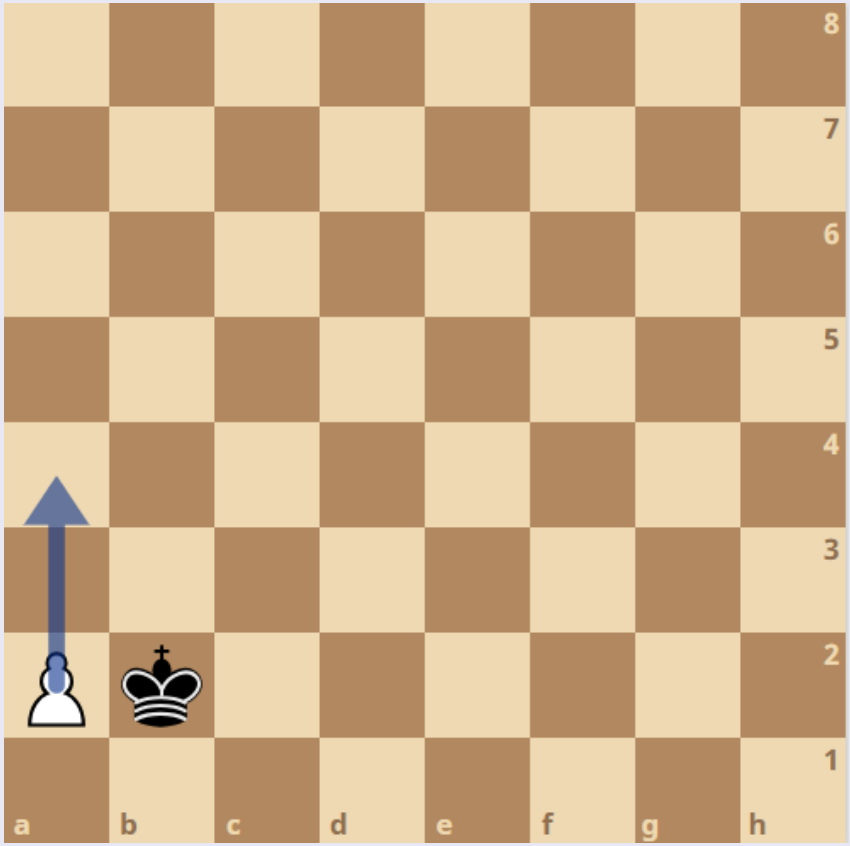


A. Final de Peón vs Rey

Existen muchos casos “tricky”.
Un caso de prueba es un tablero 8x8 con todas las posiciones legales.
Veamos algunos...

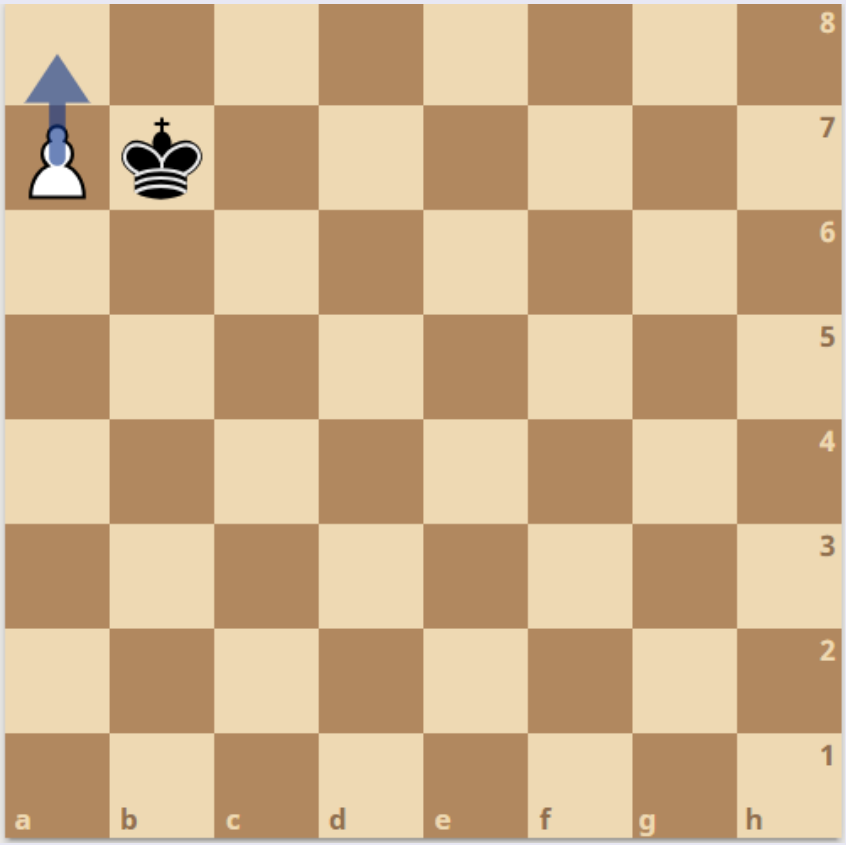
A. Final de Peón vs Rey

Caso peón blanco mueve 2 veces..



A. Final de Peón vs Rey

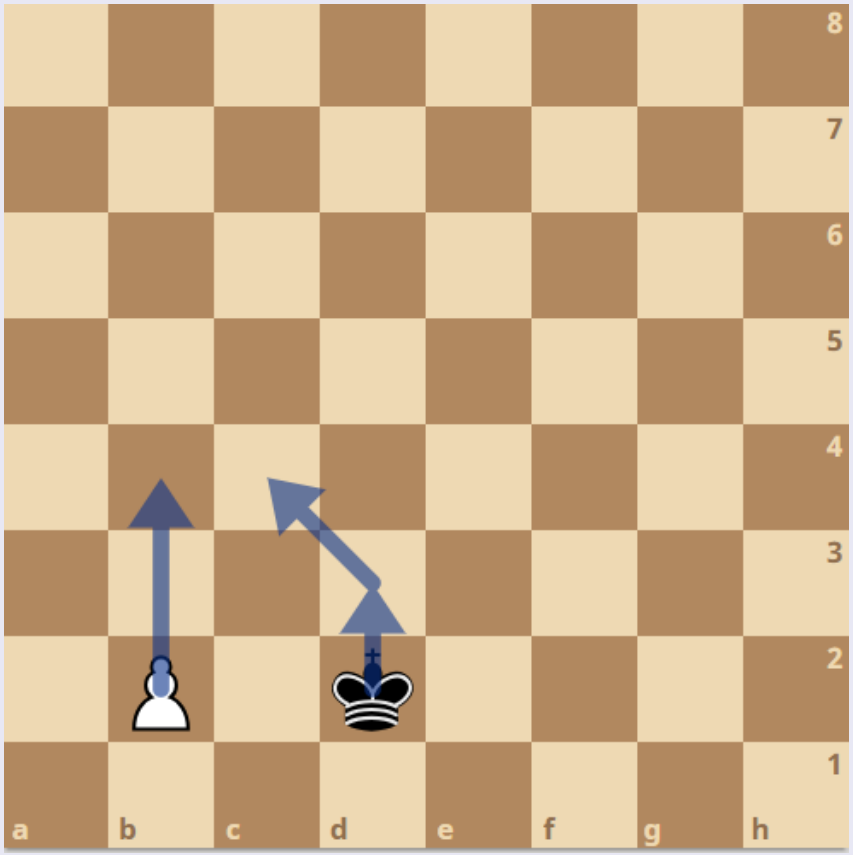
Caso de coronar y justo come..



- Se dice que un peón corona cuando llega a la última fila del tablero. En este punto se puede convertir en otra pieza a su elección (a excepción de rey o peón). Ten en cuenta que si llega a la última fila y corona, el siguiente turno es del rival, que si está al lado, puede capturar a la pieza.

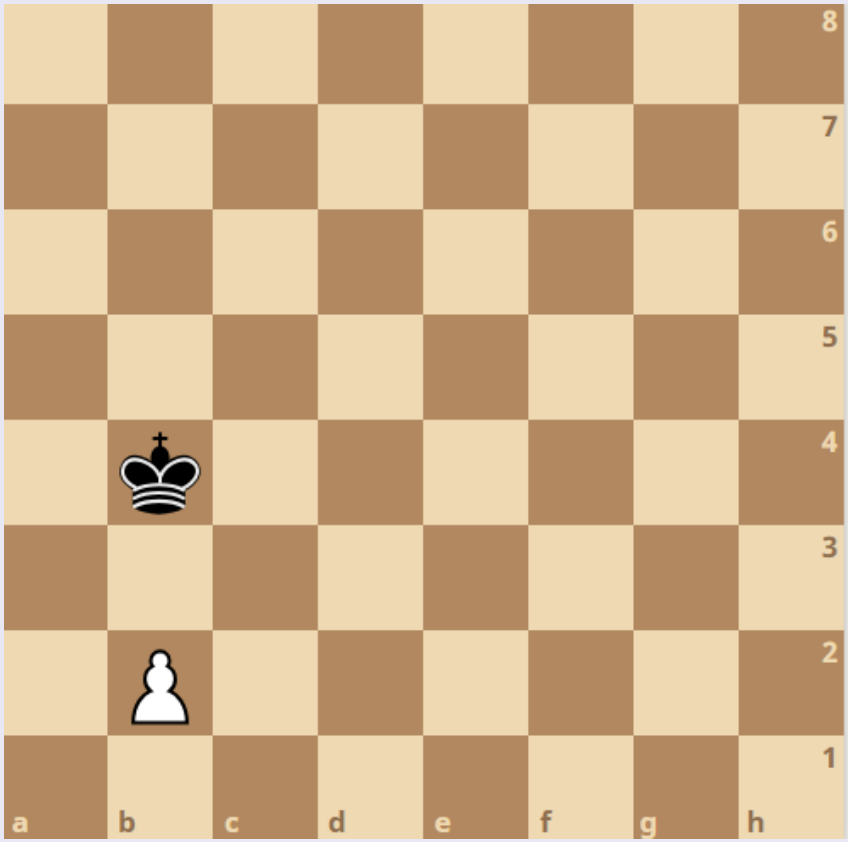
A. Final de Peón vs Rey

Caso tricky preguntado, juegan negras..



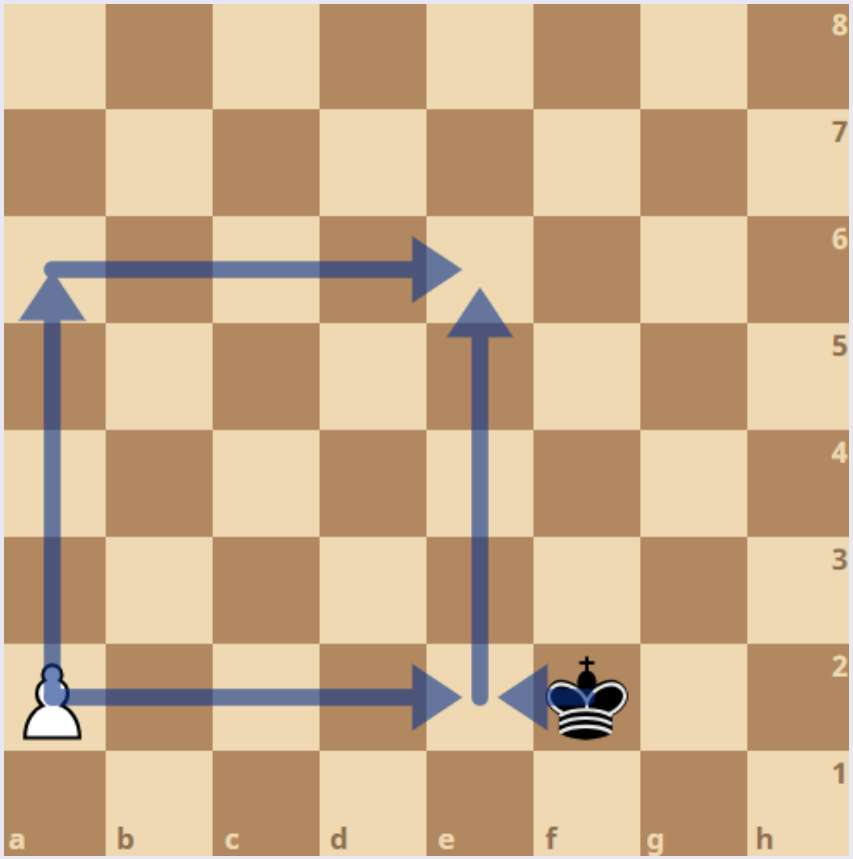
A. Final de Peón vs Rey

Caso tricky preguntado, juegan blancas..



A. Final de Peón vs Rey

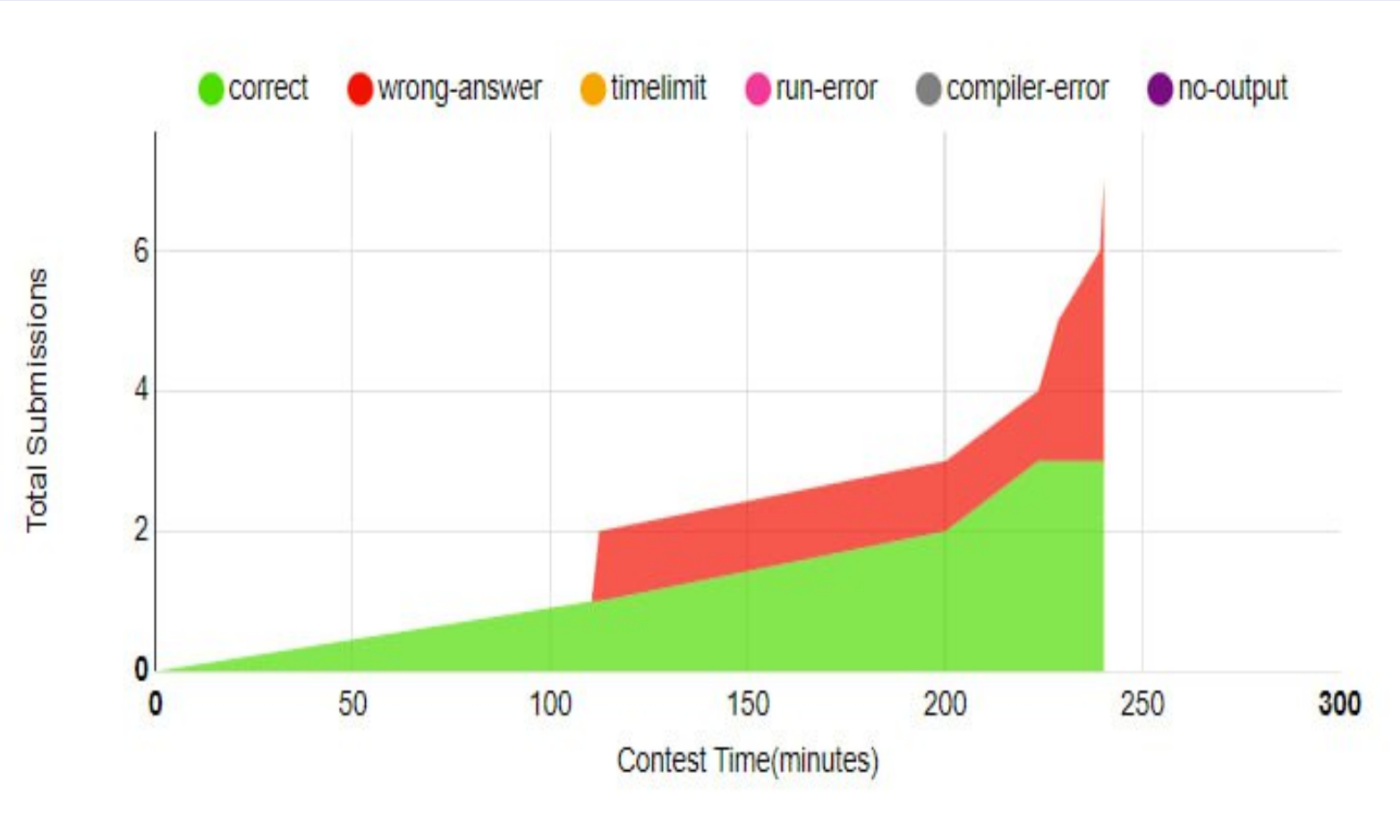
Caso tricky preguntado, juegan negras..



H. Mini tetris

Envíos	Válidos	% éxito
6	3	50 %

H. Mini tetris



H. Mini tetris

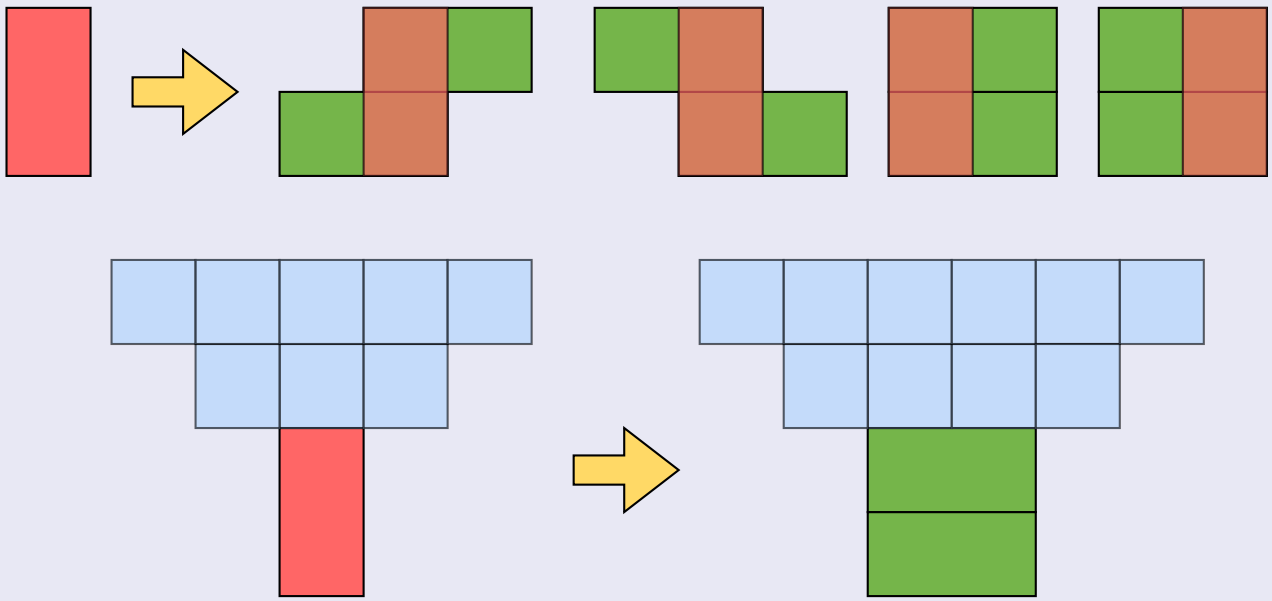
¿Cuál es la torre de bloques que puede coger mayor valor de monedas?
Normas del juego:

- El primer bloque lo puedes colocar donde quieras (apoyado en el fondo, los bloques caen como en el tetris).
- Cada bloque debe ir encima del último bloque colocado
- Puedes utilizar todos los bloques que quieras hasta que la torre llegue hasta arriba.

Veamos que tenemos que tener en cuenta:

H. Mini tetris

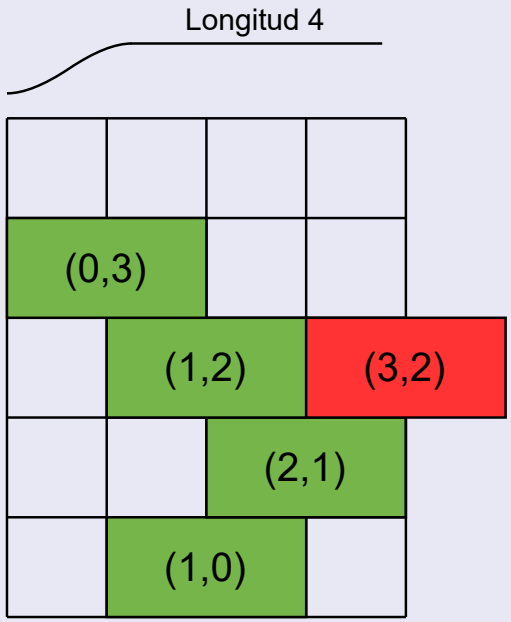
Es inútil colocar bloques verticales:



No tenemos restricciones sobre minimizar el número de bloques, unicamente maximizar el valor!

H. Mini tetris

Colocar un bloque es fácilmente identificable por la esquina izquierda (o derecha) del mismo:



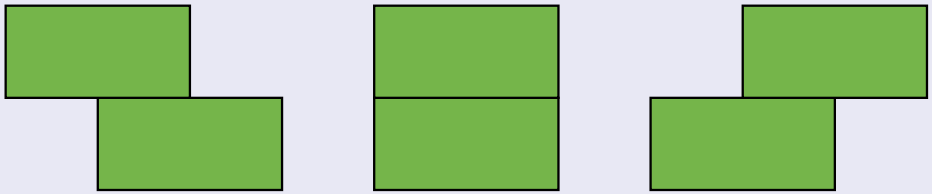
Si la anchura del escenario de juego es H, la coordenada x de los bloques no puede super $H-1$

$$x_i < H-1 \;\; o \;\; x_i \leq H-2$$

Sin embargo, al equivocarnos en esta restricción, con la natural que es $x_i < H$ no obtendriamos WA (cuidado con el RTE) ya que una buena solución no se saldría de los limites al maximizar el valor de las monedas.

H. Mini tetris

Solución(?): Probamos a colocar todas las bases posibles, a partir de ahí jugamos creando una torre, por cada bloque colocado hay 3 posibilidades, colocar el siguiente un paso a la izquierda, justamente encima, o un paso a la derecha:



Si tenemos una función $solve(x, y)$ que nos da el valor máximo obtenible de la torre que comienza con el bloque (x, y) se tiene que:

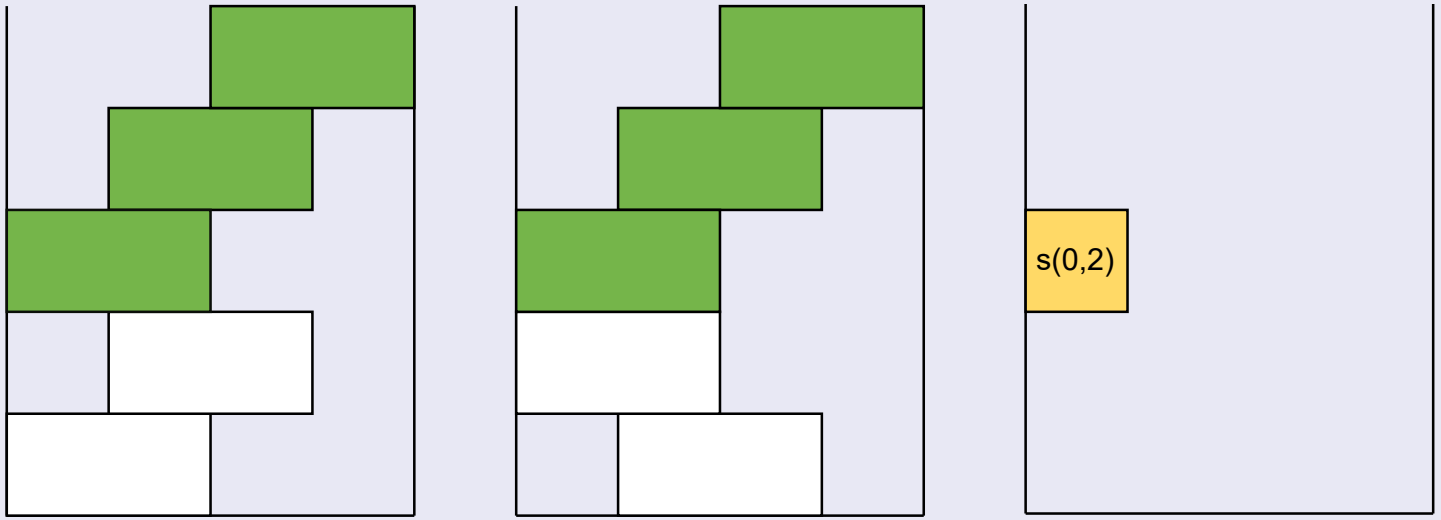
$$solve(x, y) = max\{solve(x-1, y+1), solve(x, y+1), solve(x+1, y+1)\}$$

$$+coins(x, y) + coins(x+1, y)$$

Backtracking -> $O(3^{WL})$ -> TLE

H. Mini tetris

Solución: Tenemos que acelerar el proceso!



Existen estados que se repiten en el backtracking -> **Programación Dinámica**

Usando la estructura de memo solo debemos recorrer cada estado 1 vez.

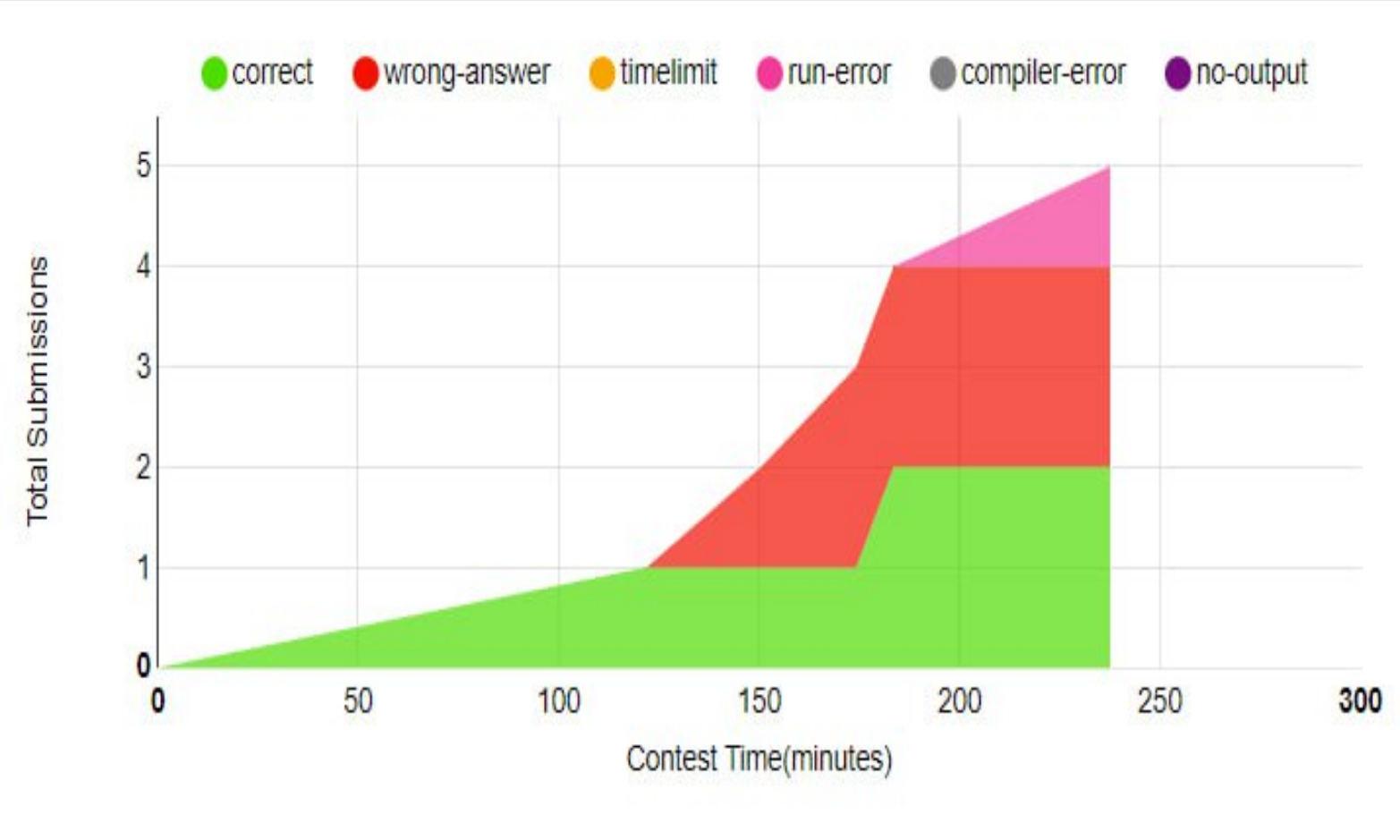
$O(WL)$ -> **AC**

CUIDADO: Almacenar las monedas en un mapa en vez de en un array -> **TLE**

B. Huelga de Agricultores

Envíos	Válidos	% éxito
5	2	40 %

B. Huelga de Agricultores



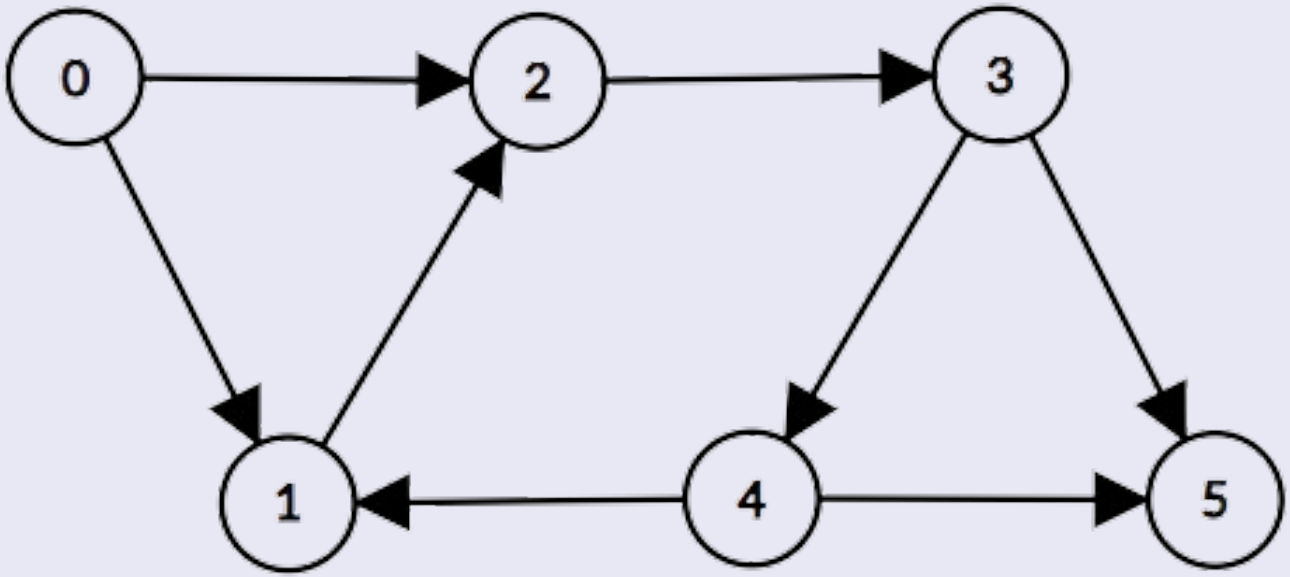
B. Huelga de Agricultores

Dado un grafo dirigido, calcular el mínimo número de aristas a bloquear.

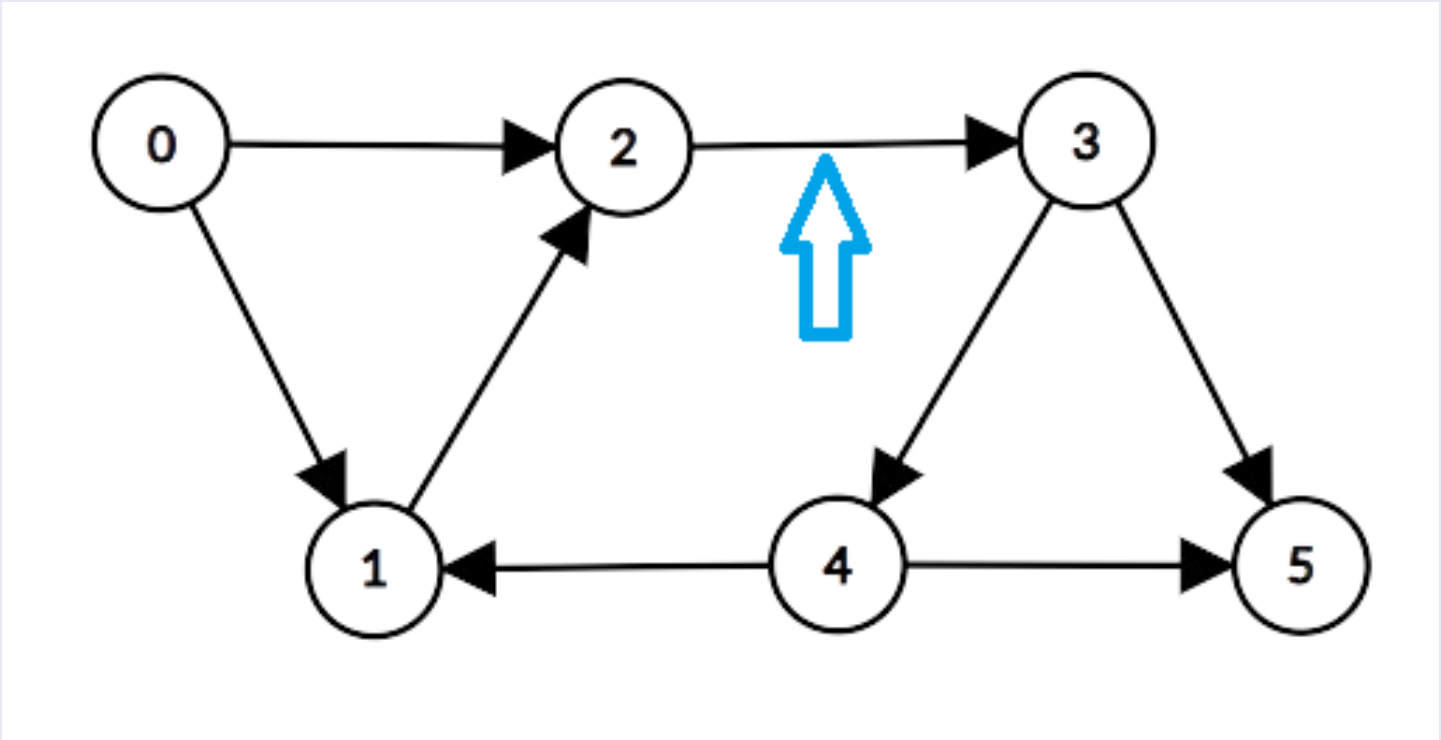
B. Huelga de Agricultores

Problema clásico de min-cut, única dificultad ¿cómo devolver las aristas? Existen soluciones múltiples (al final puedes ordenar la salida aleatoriamente o tener varias).

B. Huelga de Agricultores



B. Huelga de Agricultores



B. Huelga de Agricultores

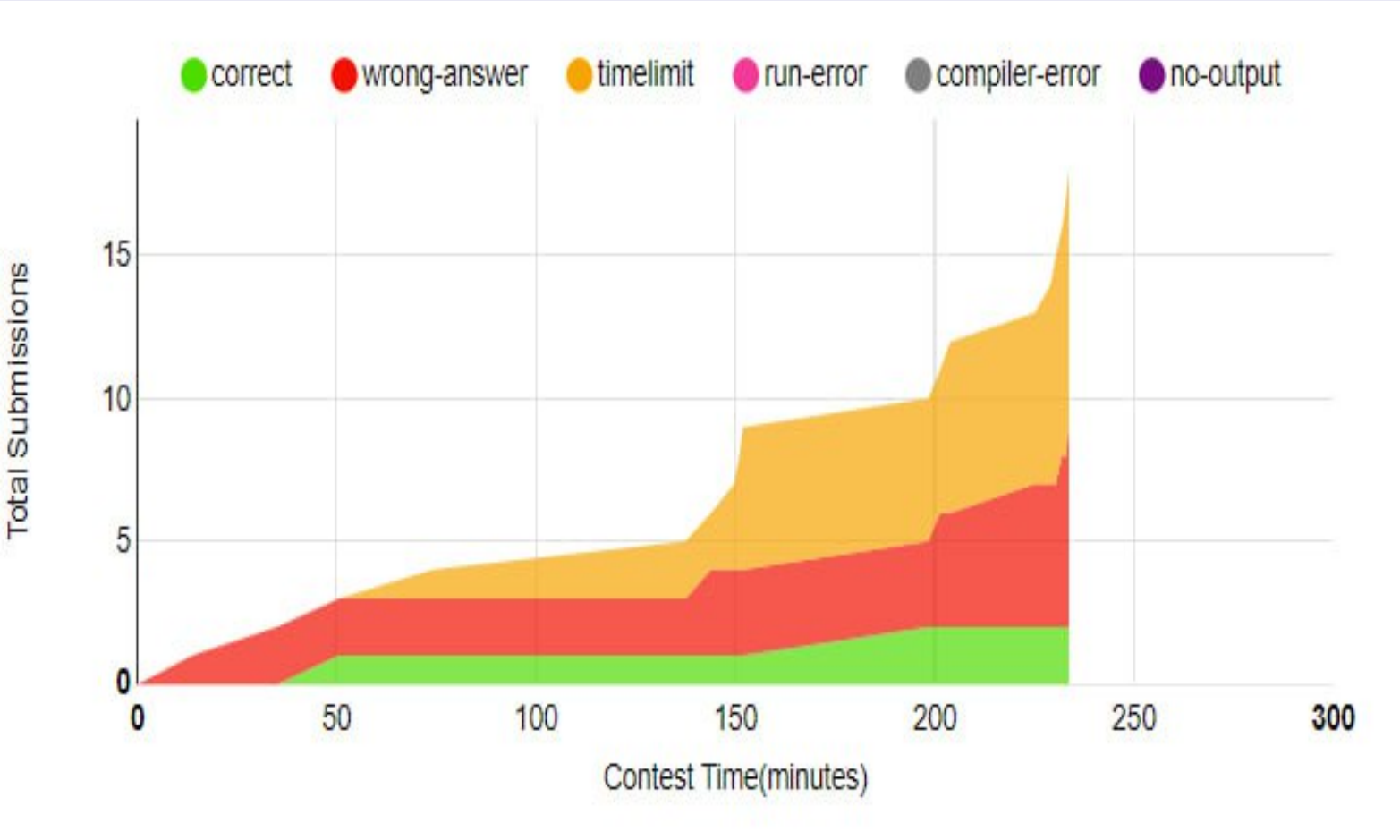
Para determinar las aristas, Sea A el conjunto de todos los nodos alcanzables desde el punto de partida y B es el conjunto de todos los nodos no alcanzables desde el punto de partida.
Ahora, el corte mínimo está formado exactamente por todas las aristas que satisfacen las siguientes propiedades:

- La arista era una de las aristas originales.
- La arista parte de un nodo de A.
- La arista termina en un nodo de B.
- La arista tiene peso cero después de ejecutar el algoritmo de flujo.

F. Follow the Lidar

Envíos	Válidos	% éxito
18	2	11 %

F. Follow the Lidar



F. Follow the Lidar

Aunque hablemos de "bailesz robots y la temática esté inspirada en SLAM (Localización y Mapeo Simultaneo)...
En realidad estamos resolviendo un laberinto.



Figura: Laberinto desde arriba



Figura: Lo mismo subjetivo

Habéis estado en un laberinto? [...]

F. Follow the Lidar

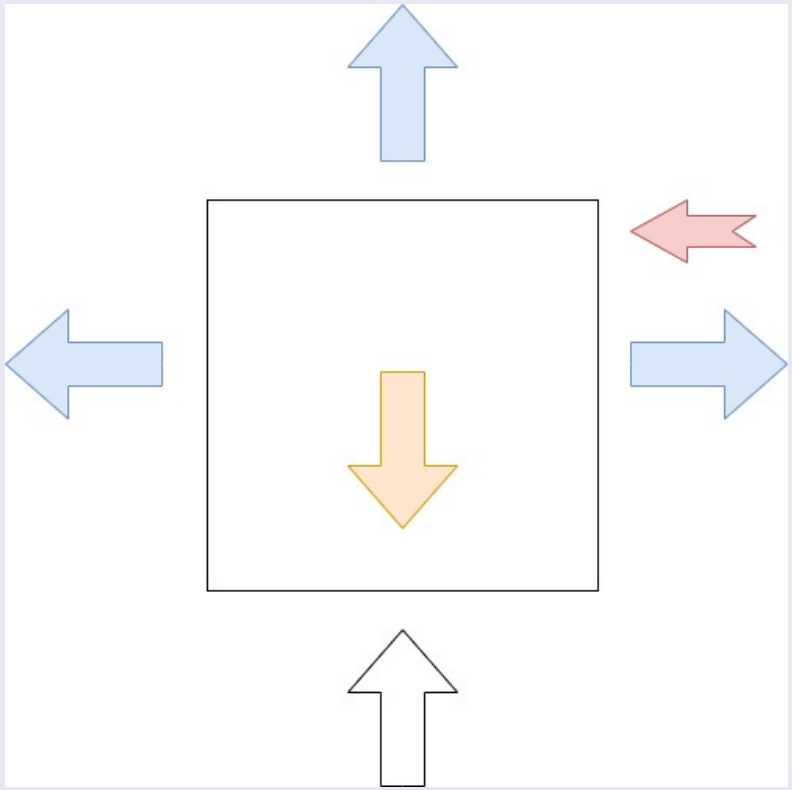
Algunas consideraciones:

- No existe una solución óptima.
- No existe una solución única.

Así pues:

- Hay que ir probando.
- Pero hay que *apuntar* el camino. Lo podemos ver al estilo clásico: hilo de Ariadna, marcar con tiza, regla de la mano derecha...
- Vamos a plantearlo como su fuese un *backtracking*.

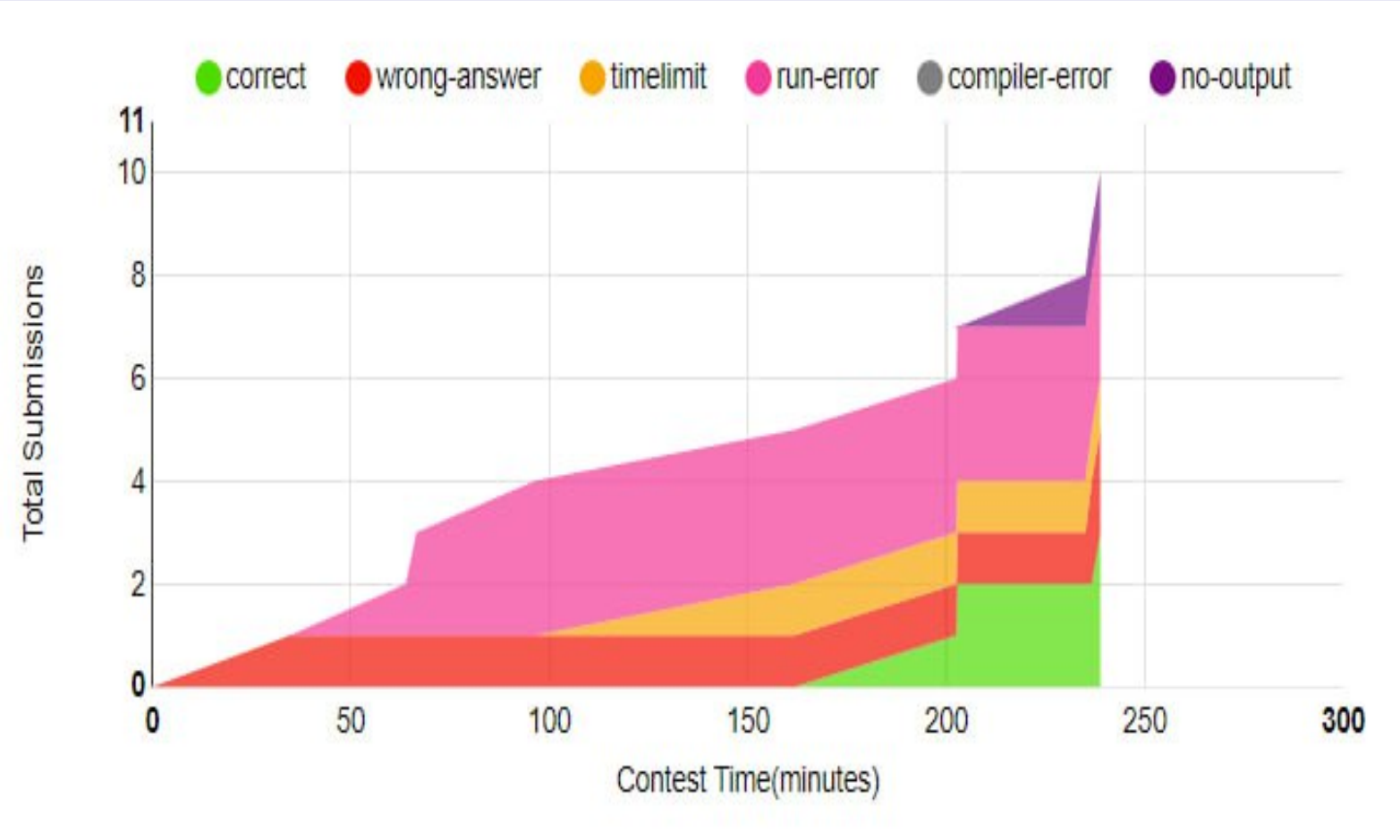
F. Follow the Lidar



C. Y sin embargo se mueve

Envíos	Válidos	% éxito
10	3	30 %

C. Y sin embargo se mueve



C. Y sin embargo se mueve

Se da un árbol (grafo sin ciclos) de n nodos y para cada arista se dan a_i y b_i.
Tras t años, la distancia entre ambos será $a_i + t \cdot b_i$.
Se dan q preguntas de la forma: Para dos nodos u y v, ¿Mínimo número de años para que la distancia entre u y v sea $\geq w$?

C. Y sin embargo se mueve

¿Y la fórmula de la distancia? Si el camino entre u y v sigue las aristas $i_1, i_2, \ldots, i_m$, la distancia será:

$$(a_{i_1} + t \cdot b_{i_1}) + (a_{i_2} + t \cdot b_{i_2}) + \cdots + (a_{i_m} + t \cdot b_{i_m}) =$$

$$(a_{i_1} + a_{i_2} + \cdots + a_{i_m}) + t \cdot (b_{i_1} + b_{i_2} + \cdots + b_{i_m}) = d_a + t \cdot d_b$$

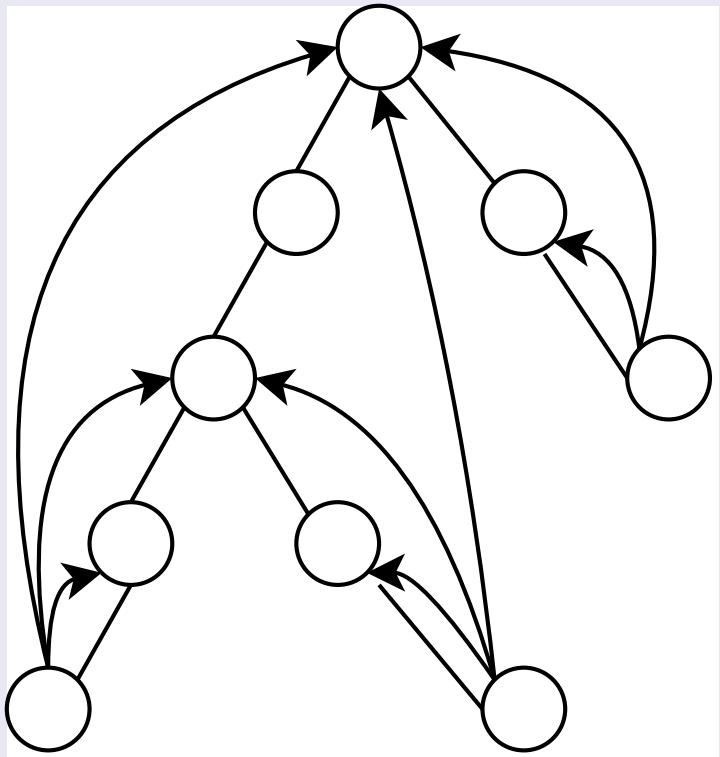
Para una pregunta entre u y v y tiempo al menos w:

$$d_a + t \cdot d_b \geq w \Rightarrow t \geq \frac{w - d_a}{d_b} \Rightarrow t \geq \left\lceil \frac{w - d_a}{d_b} \right\rceil$$

- ¿Recorrer el árbol desde u buscando v con búsqueda en anchura (BFS) o profundidad (DFS) para cada pregunta? $O(q \cdot n)$ TLE
- ¿Precalcular distancias entre cada par de nodos recorriendo n veces el árbol? $O(n^2 + q)$ RTE, MLE, demasiada memoria. (!!!)
- ¿Podemos encontrar para cada pregunta d_a y d_b lo suficientemente rápido?

C. Y sin embargo se mueve

- Pensar árbol con raíz. Precalcular para cada nodo sus ancestros potencias de 2 (1, 2, 4, 8...) $O(n \cdot \log n)$.
- Después se puede calcular distancia entre dos nodos en $O(\log n)$ saltando con potencias de 2 hasta el primer ancestro común.

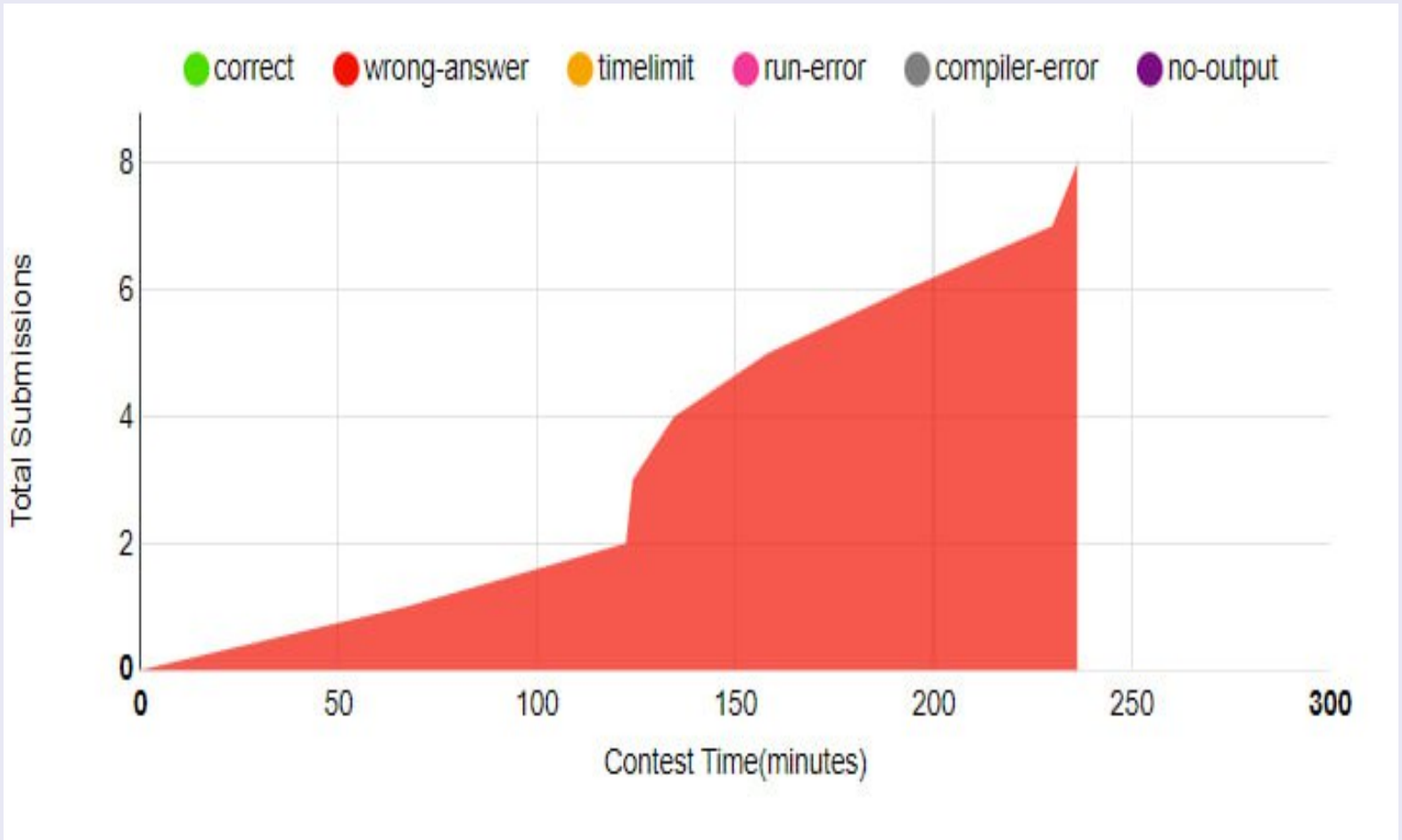


Complejidad final (precálculo + preguntas): $O(n \cdot \log n + q \cdot \log n)$

D. El problema de las 3 fuentes

Envíos	Válidos	% éxito
8	0	0 %

D. El problema de las 3 fuentes



D. El problema de las 3 fuentes

- Es un problema "clásico" de geometría computacional.
- Es la versión maximin del problema de la p-mediana para $n = 3$ y $p = 1$. (Colocar una tienda, con 3 fuentes).
- No se ha descubierto una algoritmo óptimo para $p > 1$.

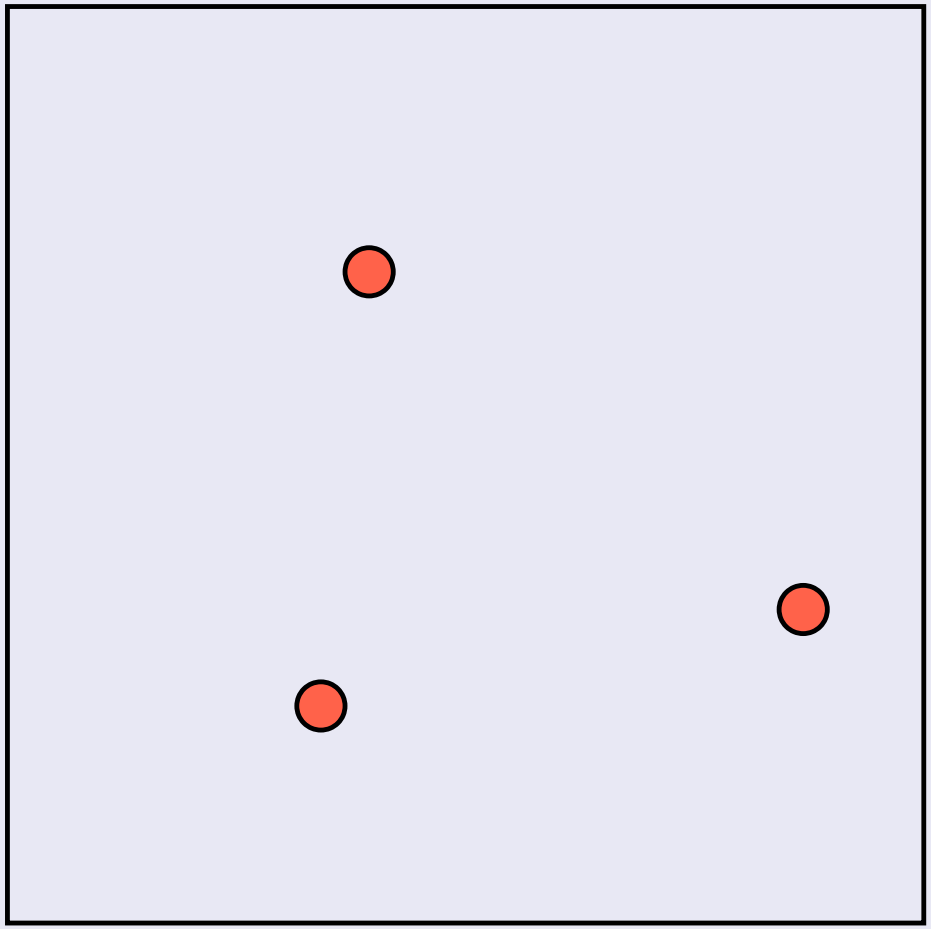
D. El problema de las 3 fuentes

PERO

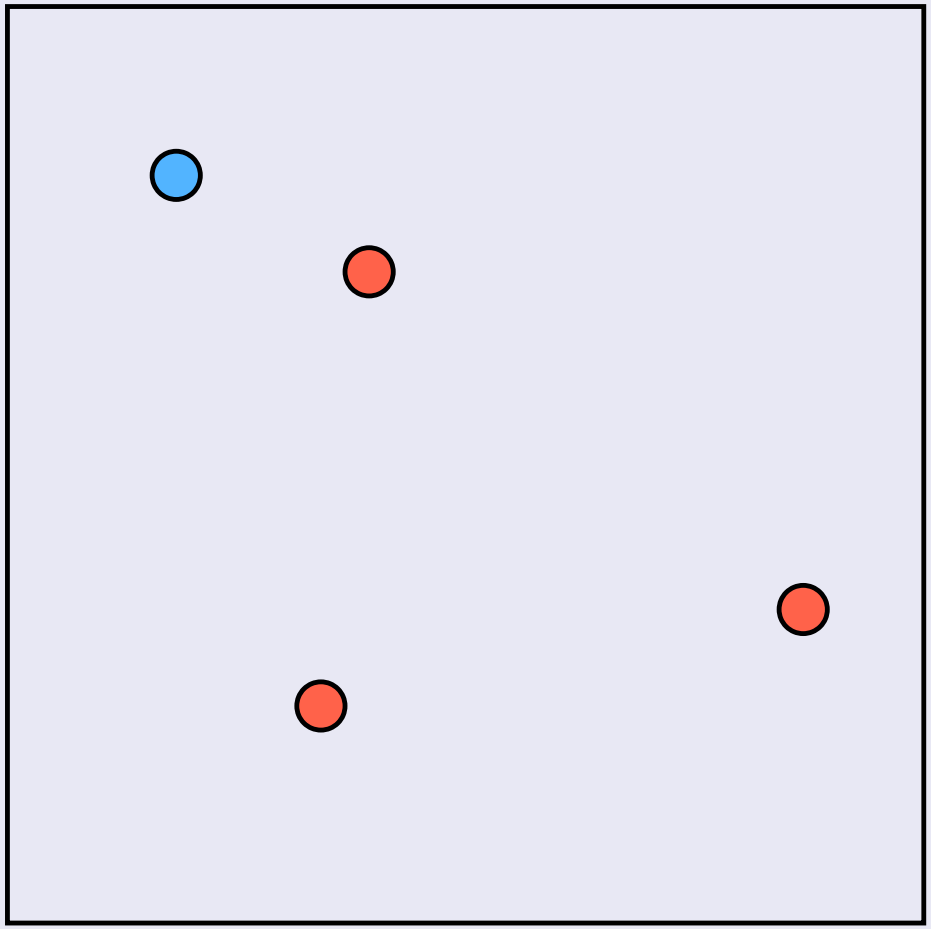
D. El problema de las 3 fuentes

La solución para $p = 1$, $n = 3$ es conocida y (...o eso creía yo...) relativamente sencilla.

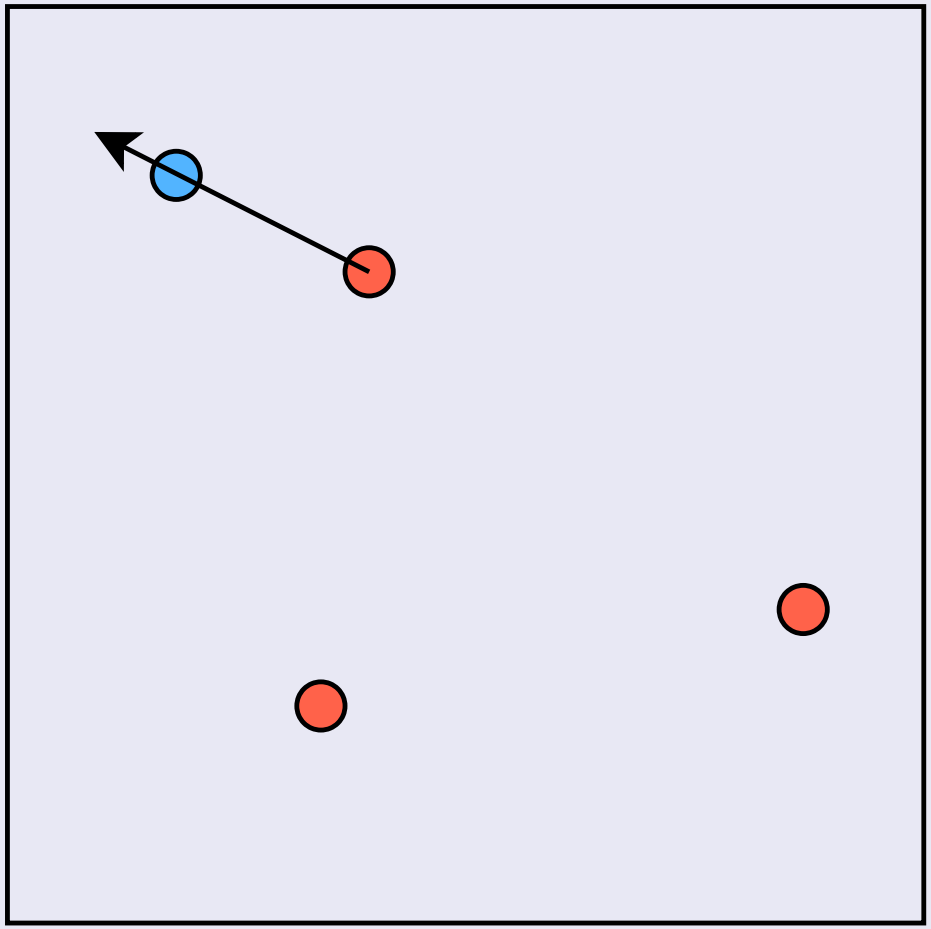
D. El problema de las 3 fuentes



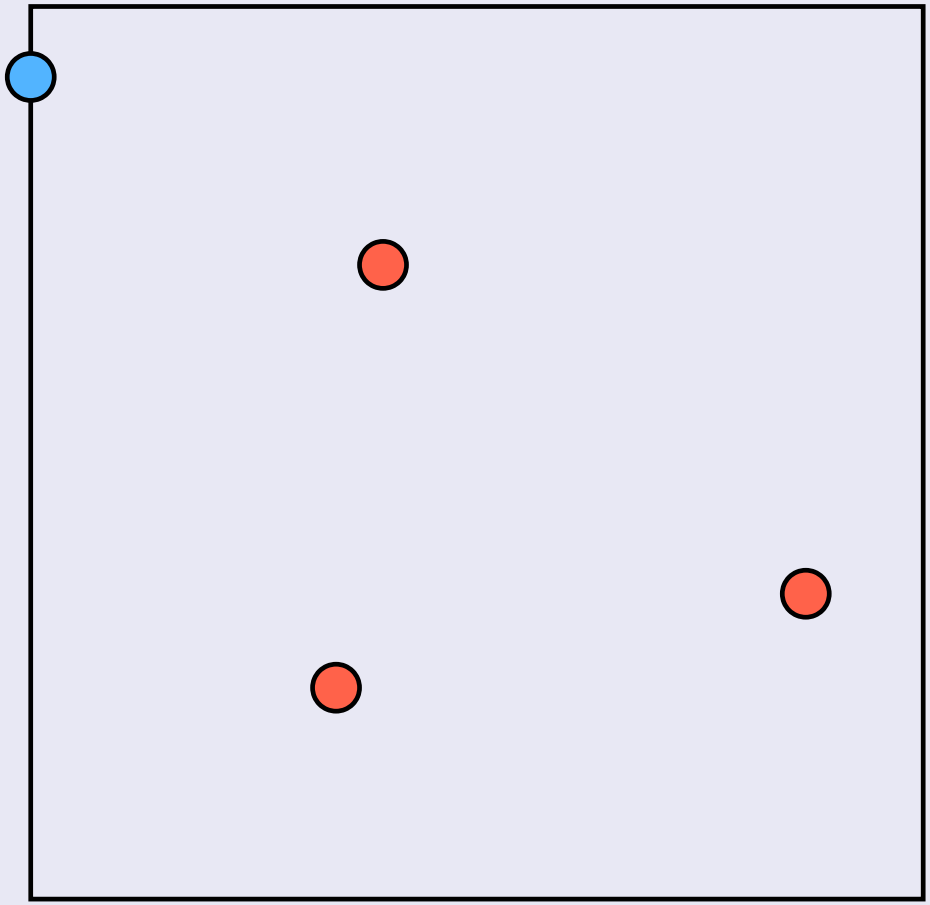
D. El problema de las 3 fuentes



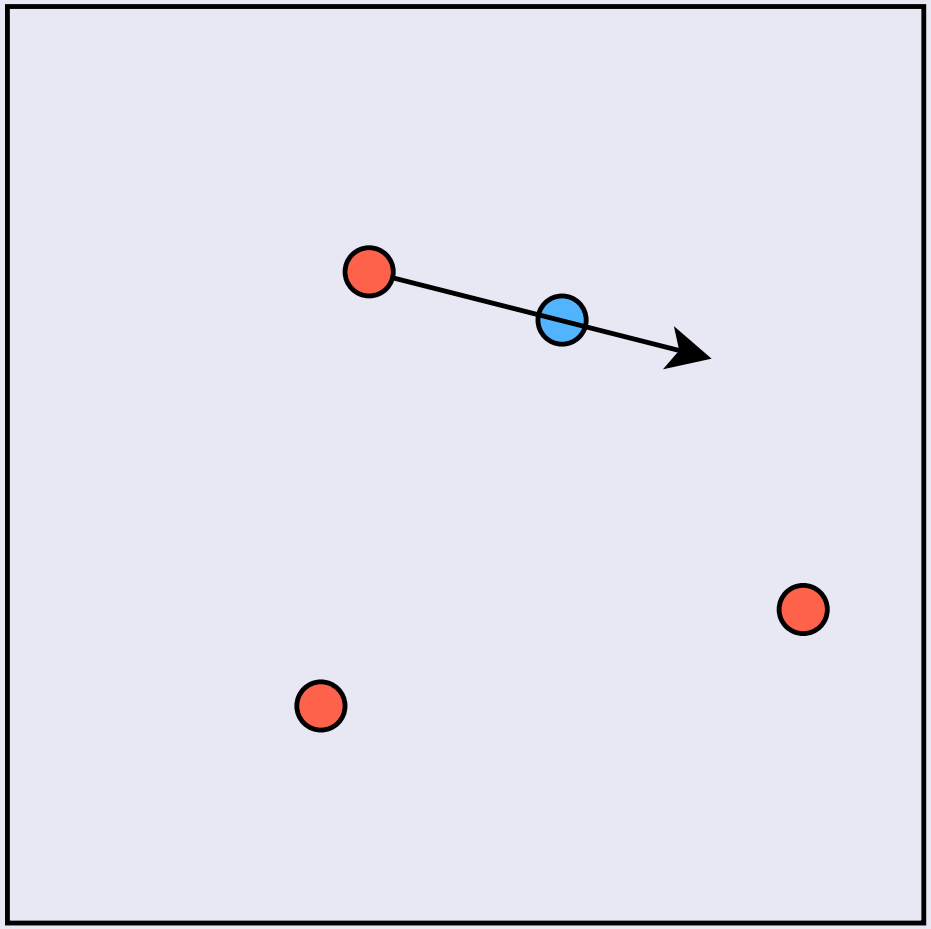
D. El problema de las 3 fuentes



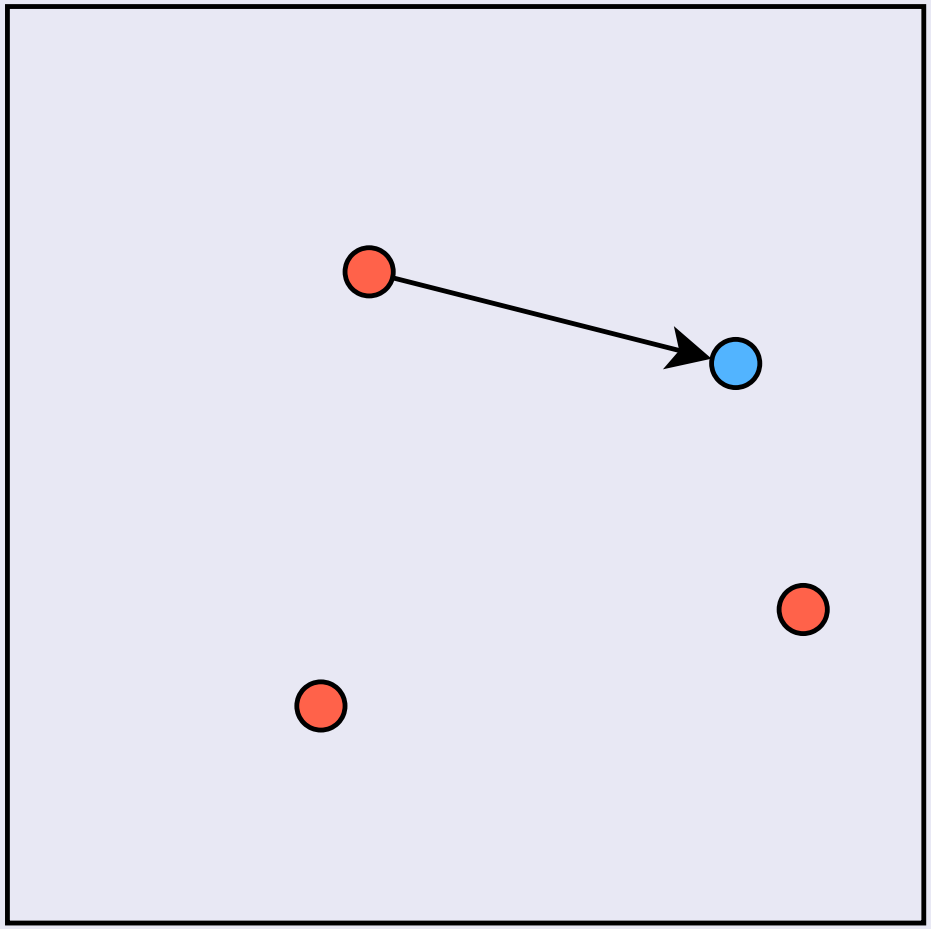
D. El problema de las 3 fuentes



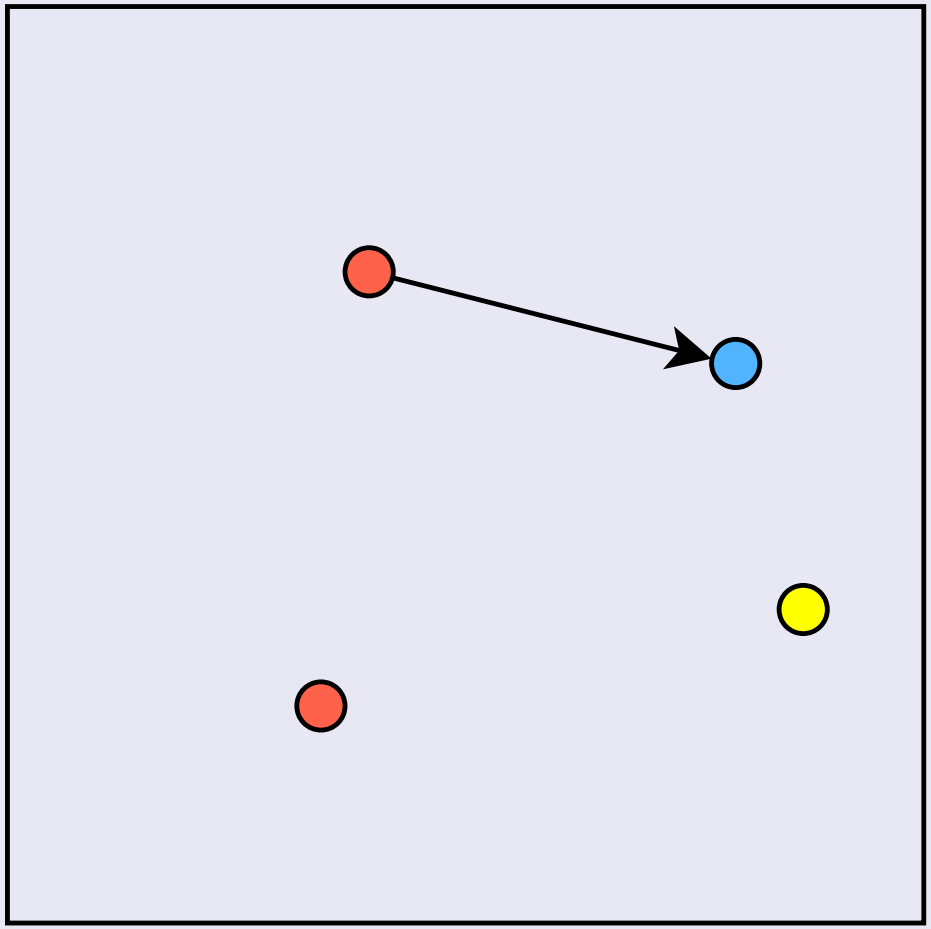
D. El problema de las 3 fuentes



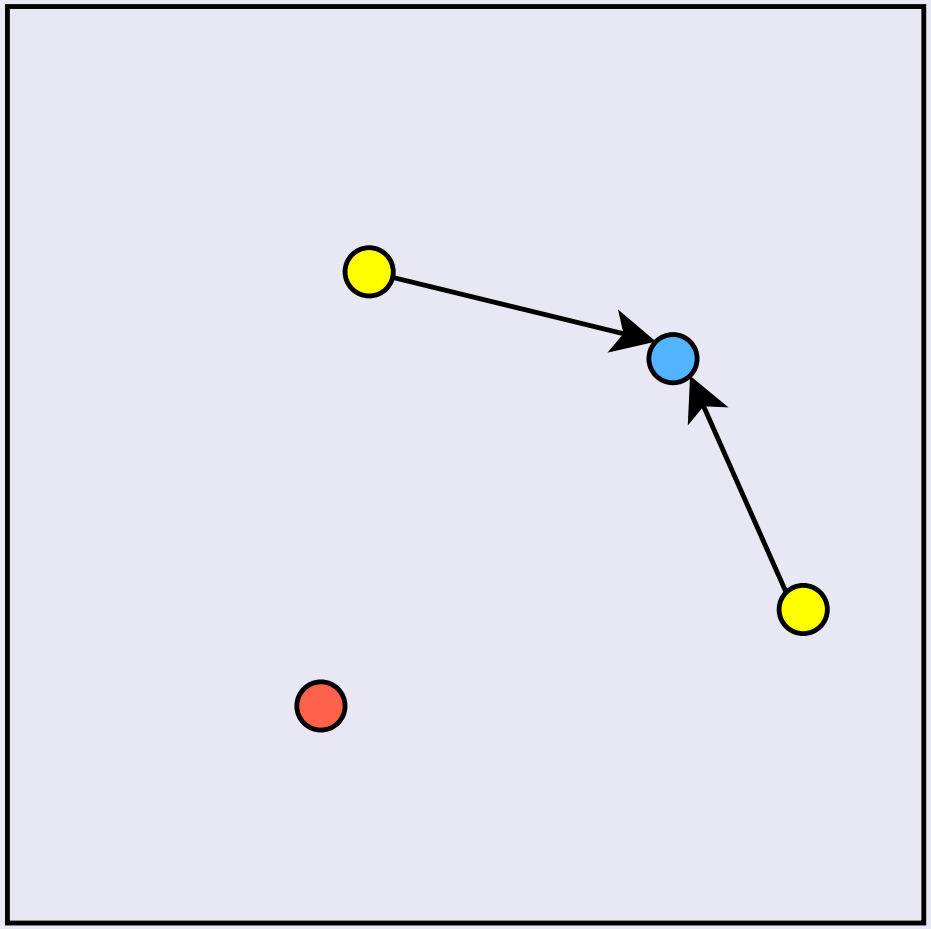
D. El problema de las 3 fuentes



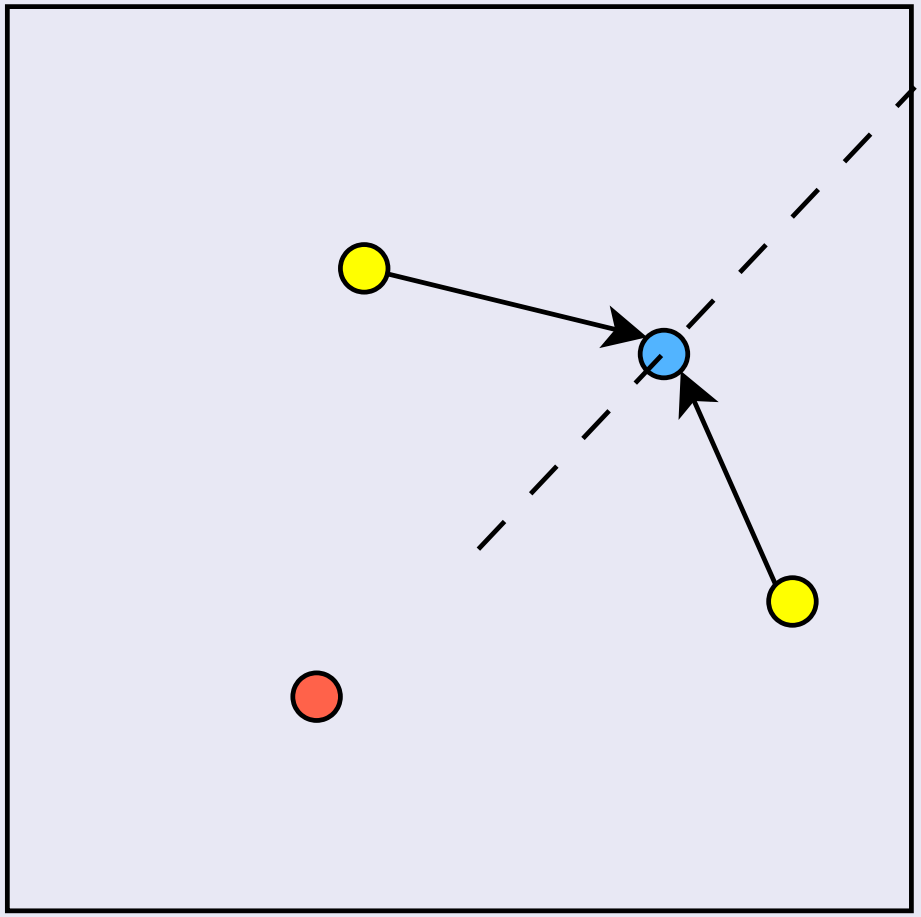
D. El problema de las 3 fuentes



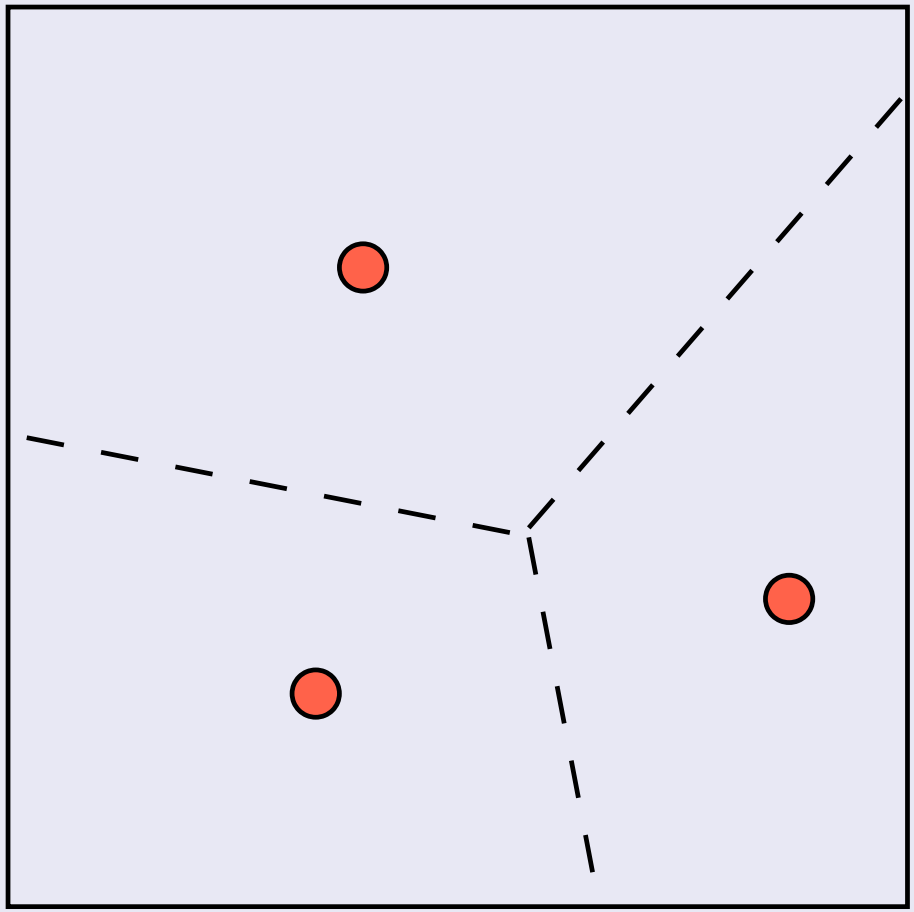
D. El problema de las 3 fuentes



D. El problema de las 3 fuentes



D. El problema de las 3 fuentes

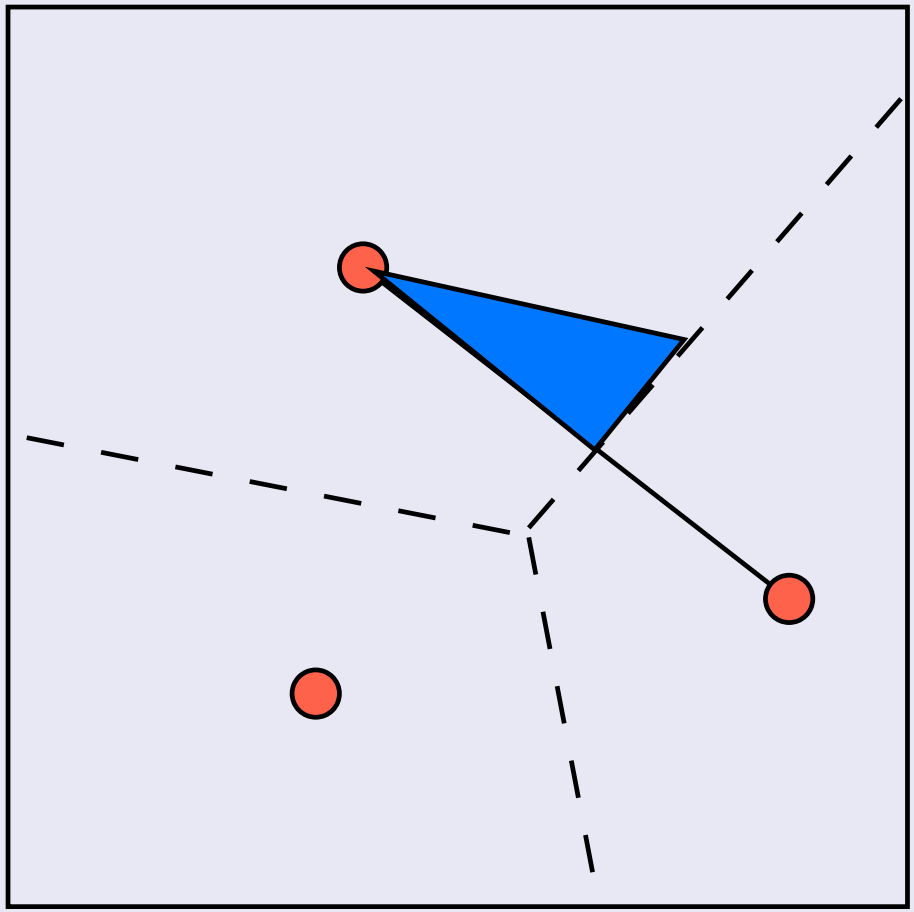


D. El problema de las 3 fuentes

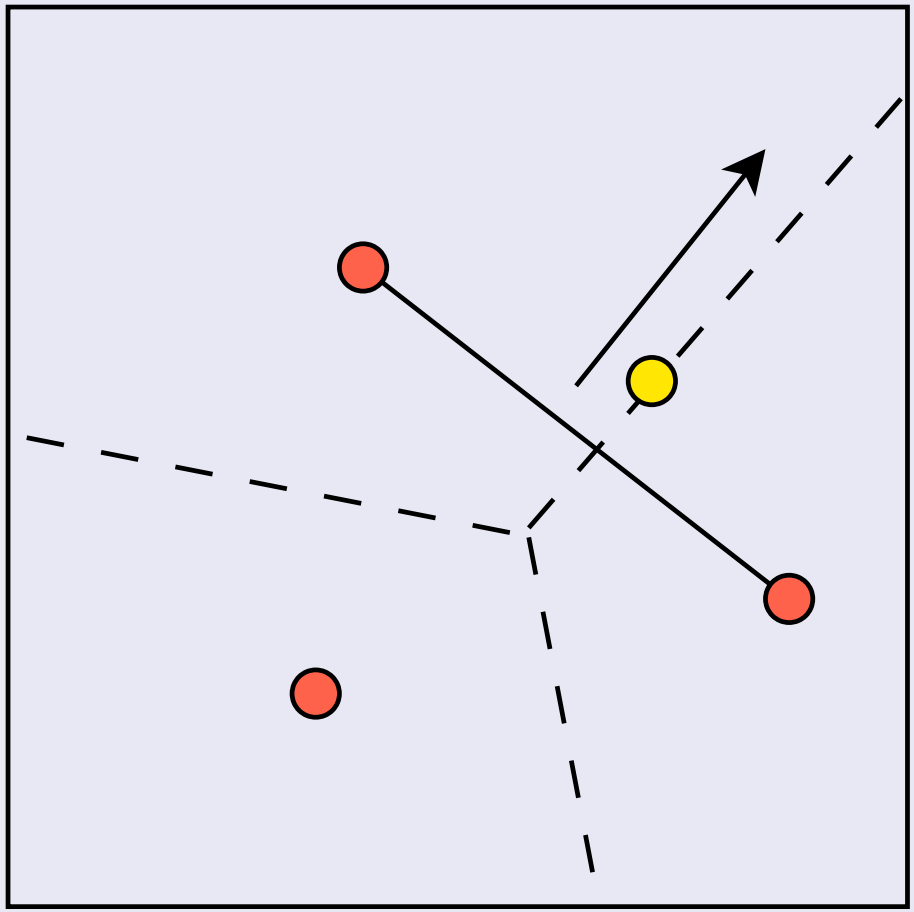
- Hemos reducido conceptualmente de el cuadrado C completo a un conjunto de rectas.
- La imagen que hemos obtenido es el diagrama de Voronoi para $n = 3$.
- ¿Necesitabais saber que era un Diagrama de Voronoi? Realmente.... no.

¿Cómo seguimos reduciendo el problema?

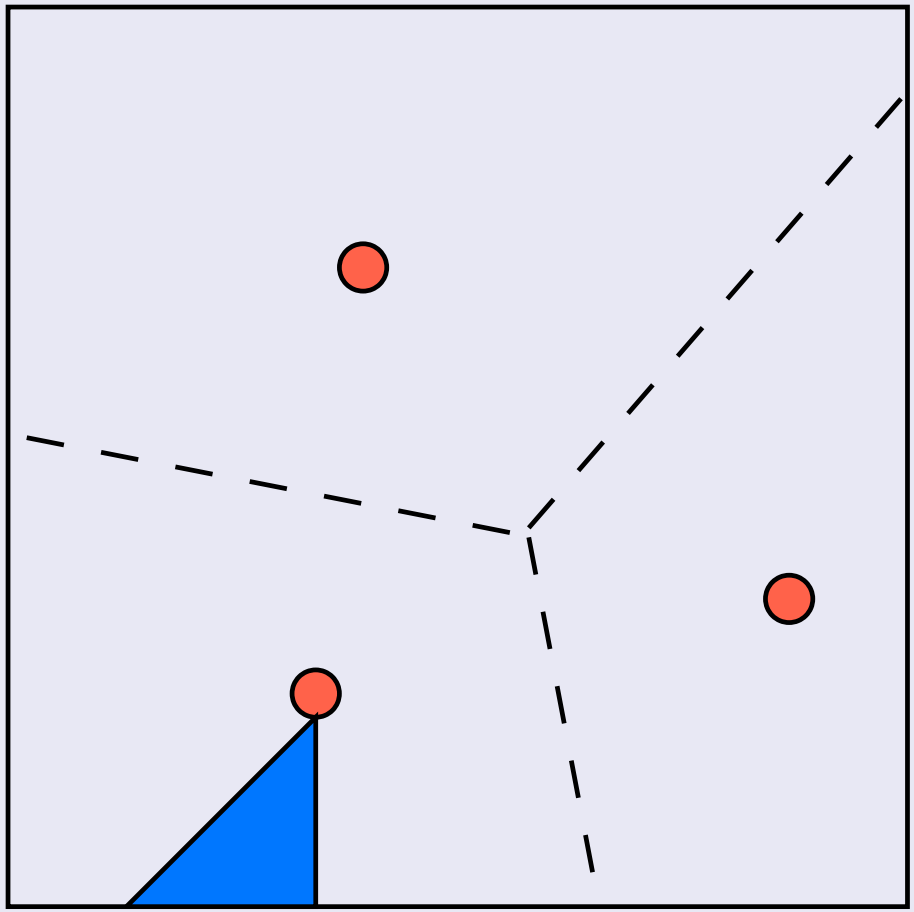
D. El problema de las 3 fuentes



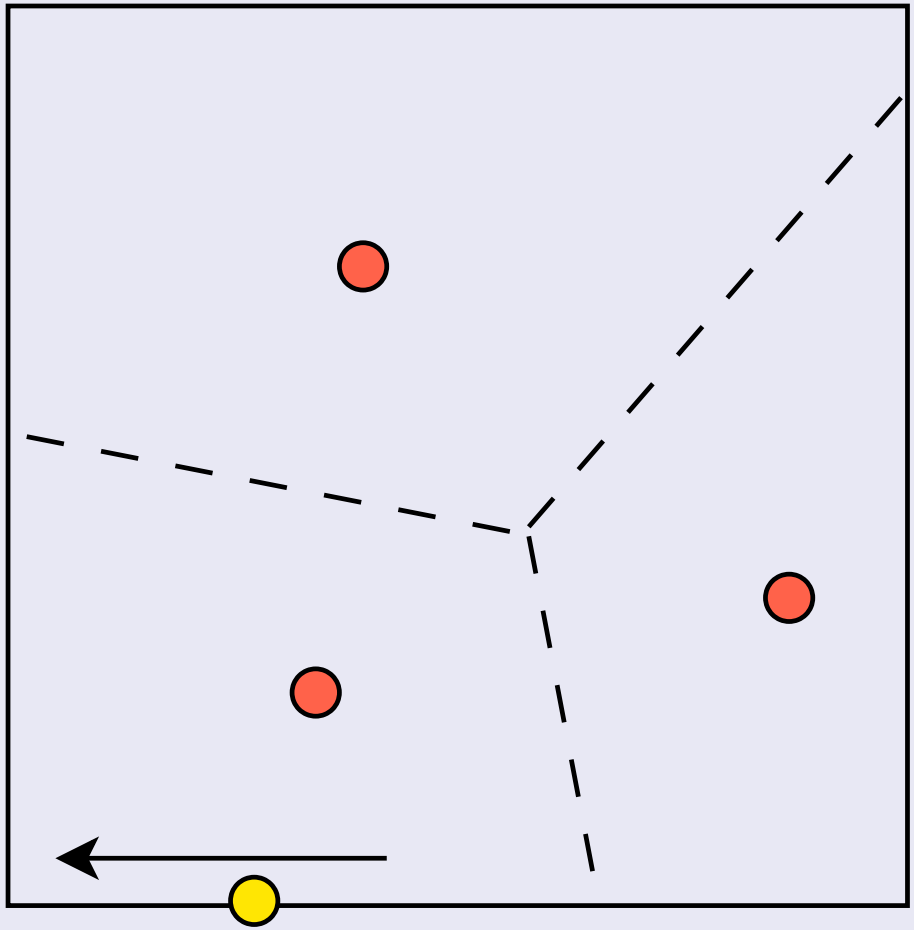
D. El problema de las 3 fuentes



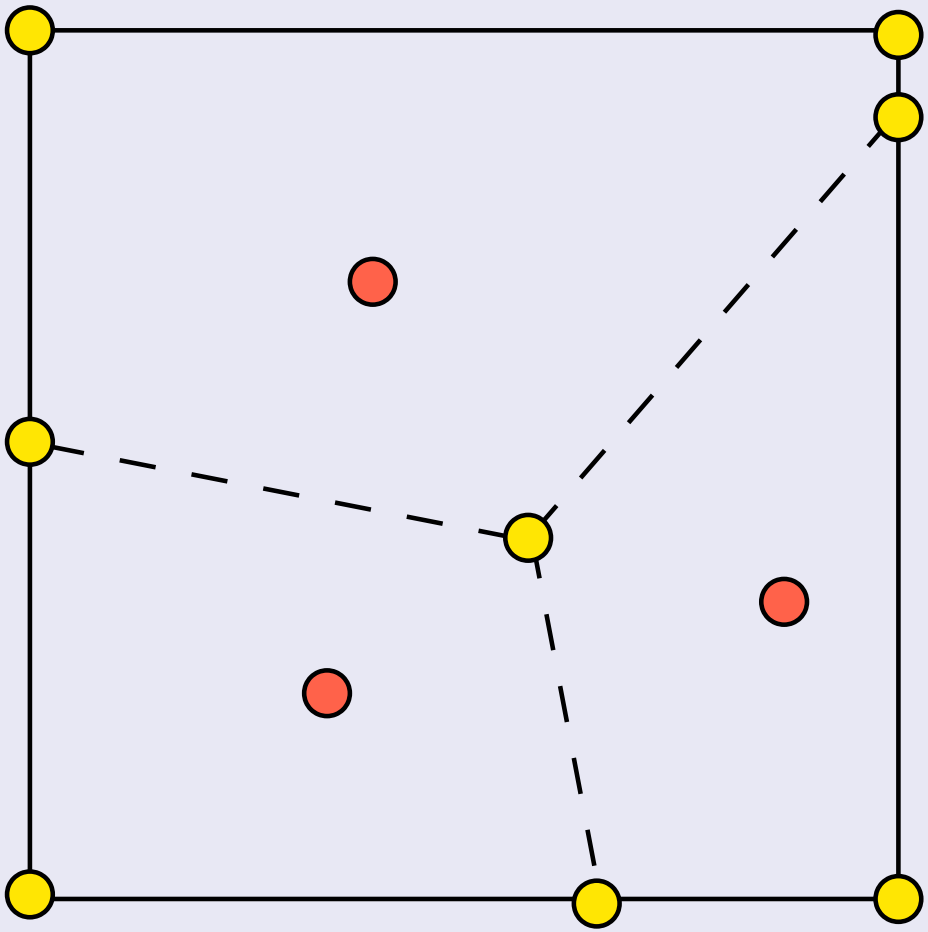
D. El problema de las 3 fuentes



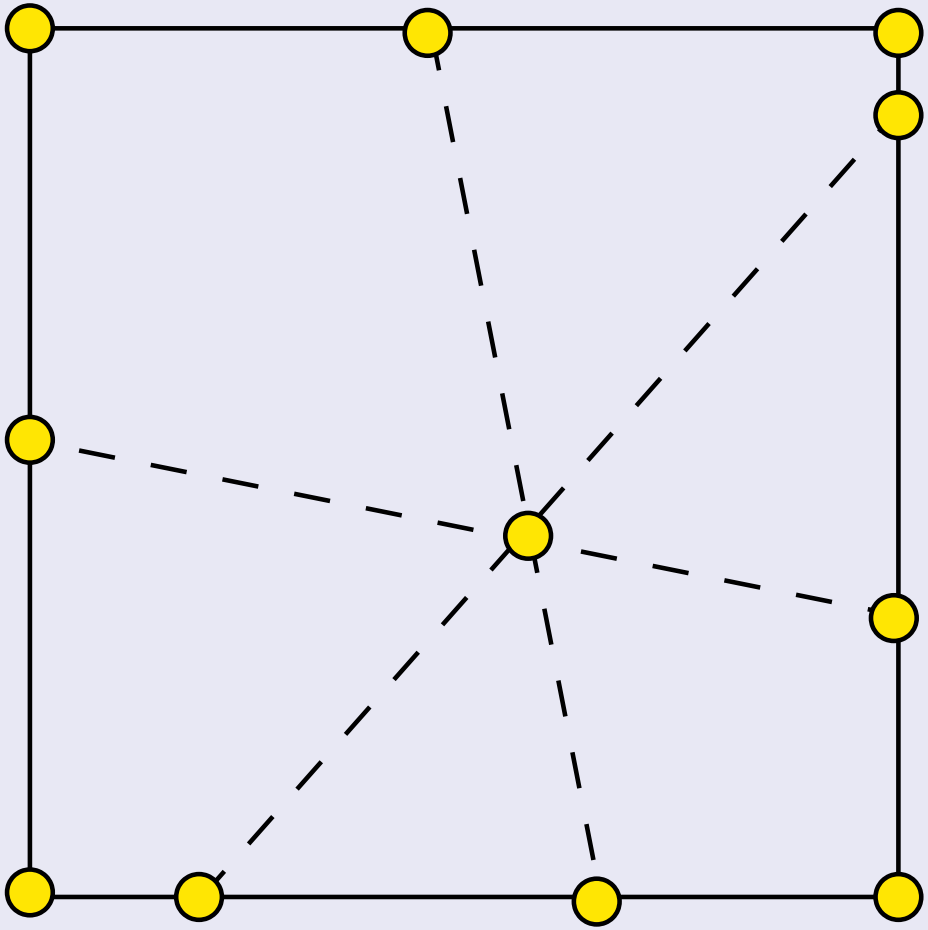
D. El problema de las 3 fuentes



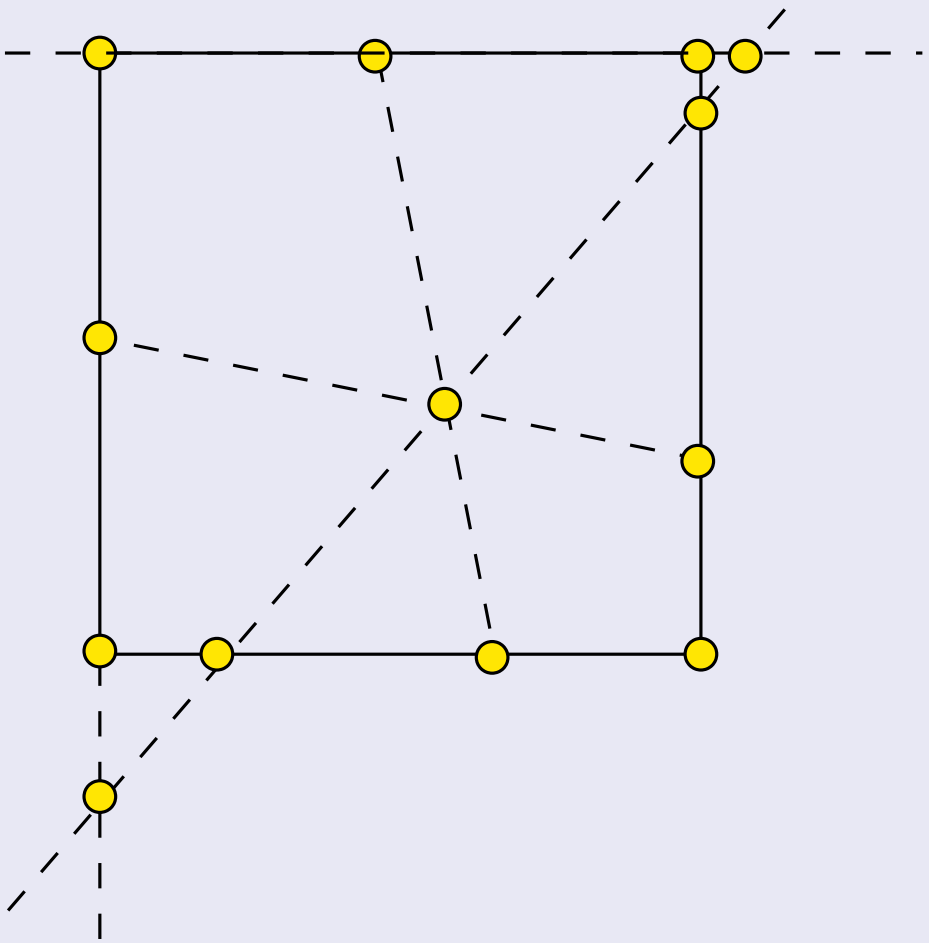
D. El problema de las 3 fuentes



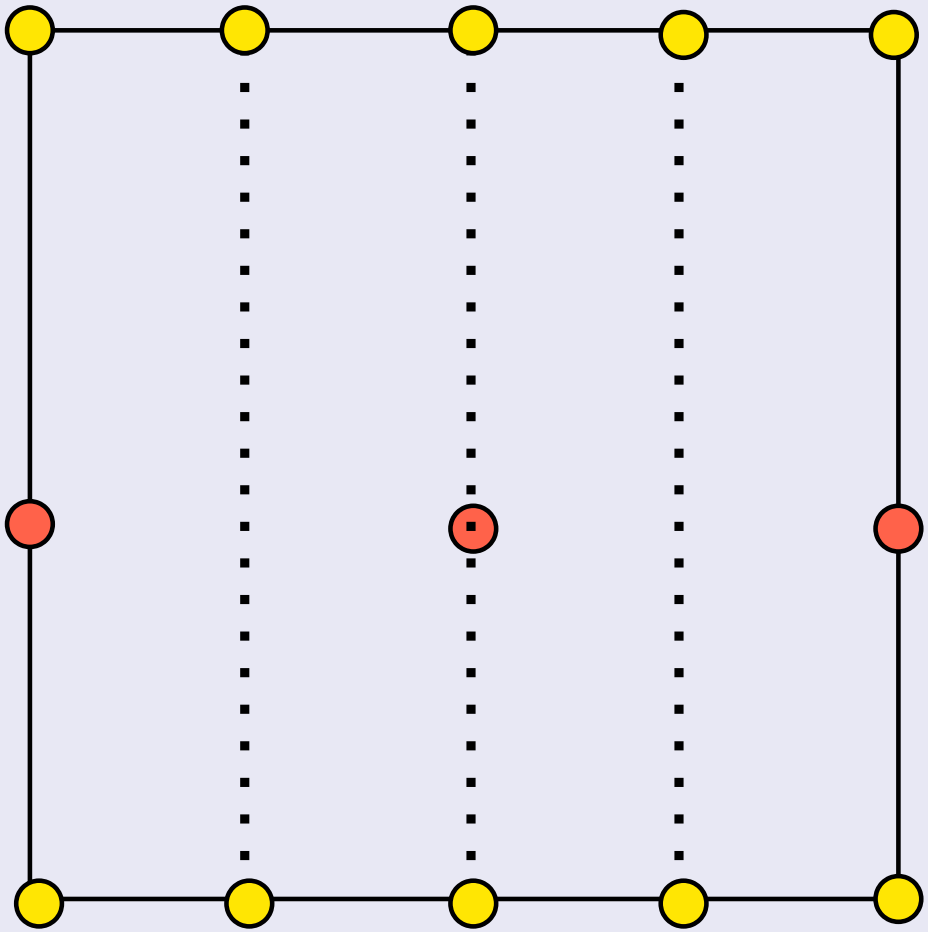
D. El problema de las 3 fuentes



D. El problema de las 3 fuentes



D. El problema de las 3 fuentes



D. El problema de las 3 fuentes

Solución final:

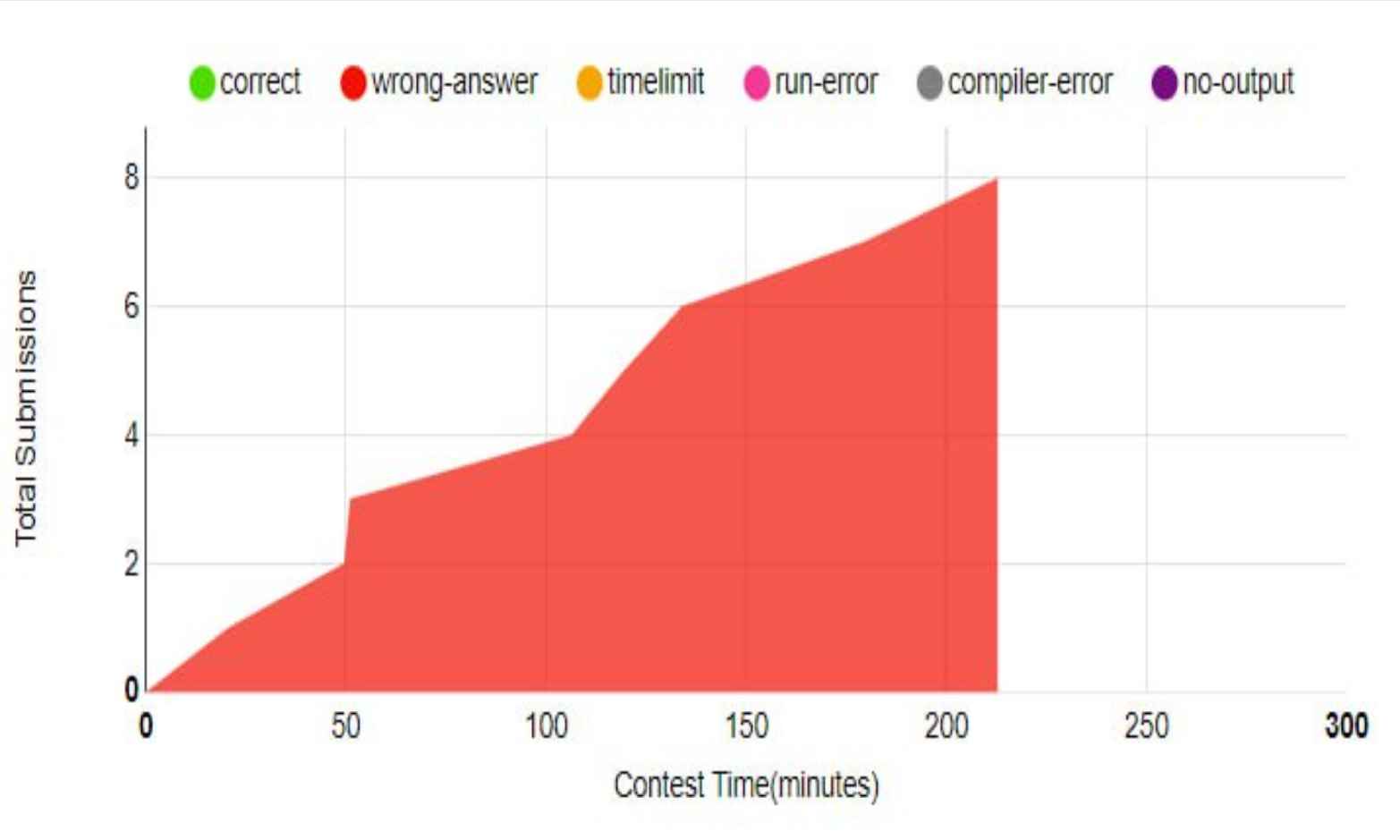
- Obten las 4 rectas que conforman el cuadrado.
  - $y = 0$
  - $y = 100$
  - $x = 0$
  - $x = 100$
- Obten las 3 mediatrices entre los pares de fuentes.
- Tenemos un total de 7 rectas
- Calcula todos los puntos de corte entre ellas
- Aquel punto que esté dentro del cuadrado y tenga el mayor valor de sequedad (mayor distancia a la fuente más cercana) es la solución.

Cuidado: Si dos rectas son paralelas no se cortan, ignoradlas.

I. El calcentín mágico

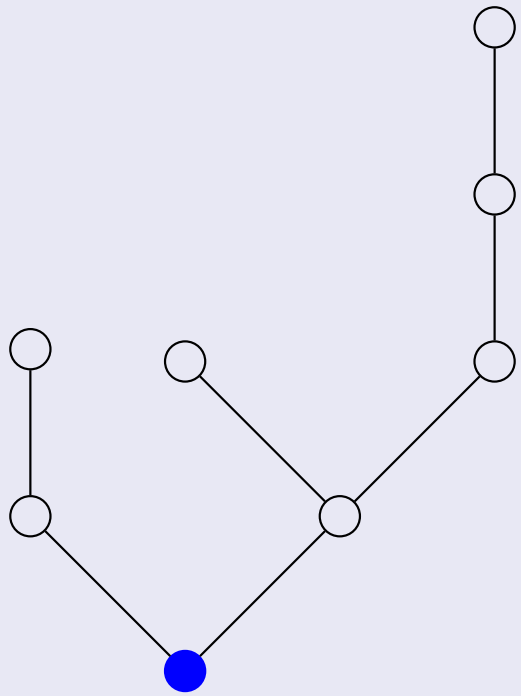
Envíos	Válidos	% éxito
8	0	0 %

I. El calcentín mágico



I. El calcentín mágico

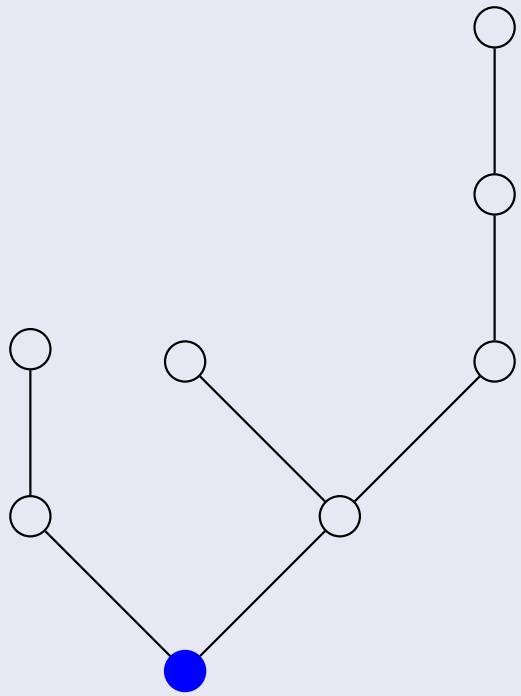
Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:



I. El calcentín mágico

Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:

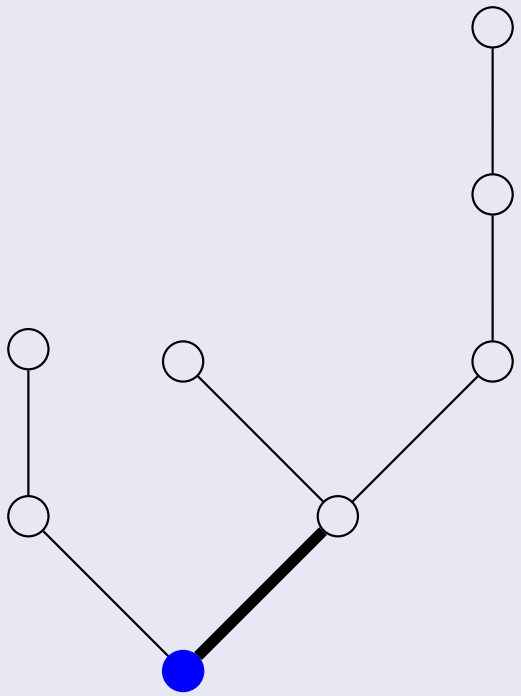
- Dos jugadores se turnan para remover aristas del árbol.



I. El calcentín mágico

Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:

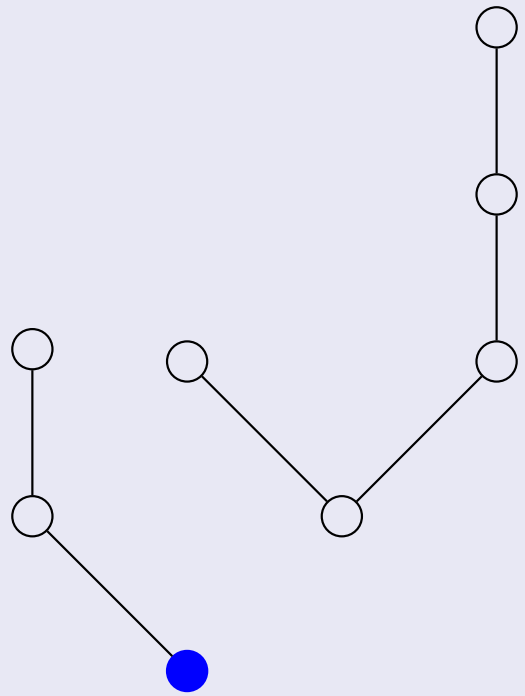
- Dos jugadores se turnan para remover aristas del árbol.
- En su turno cada jugador debe elegir exactamente una **arista** para remover.



I. El calcentín mágico

Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:

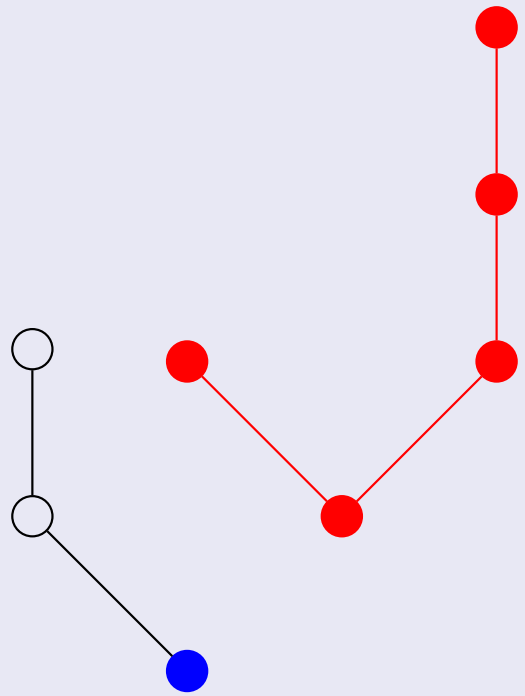
- Dos jugadores se turnan para remover aristas del árbol.
- En su turno cada jugador debe elegir exactamente una **arista** para remover.



I. El calcentín mágico

Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:

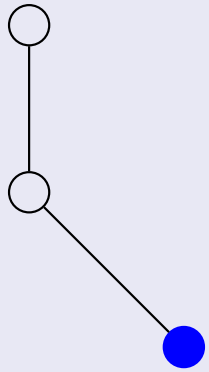
- Dos jugadores se turnan para remover aristas del árbol.
- En su turno cada jugador debe elegir exactamente una **arista** para remover.
- Cuando una arista es removida por un jugador también se elimina el subárbol que ya no queda conectado a la raíz del arbol.



I. El calcentín mágico

Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:

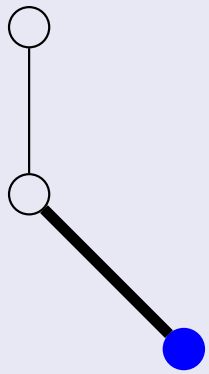
- Dos jugadores se turnan para remover aristas del árbol.
- En su turno cada jugador debe elegir exactamente una **arista** para remover.
- Cuando una arista es removida por un jugador también se elimina el subárbol que ya no queda conectado a la raíz del arbol.
- El jugador que no puede jugar pierde.



I. El calcentín mágico

Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:

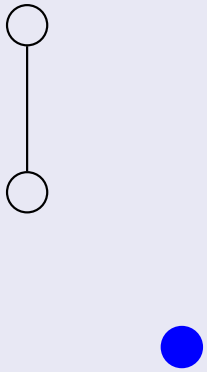
- Dos jugadores se turnan para remover aristas del árbol.
- En su turno cada jugador debe elegir exactamente una **arista** para remover.
- Cuando una arista es removida por un jugador también se elimina el subárbol que ya no queda conectado a la raíz del arbol.
- El jugador que no puede jugar pierde.



I. El calcentín mágico

Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:

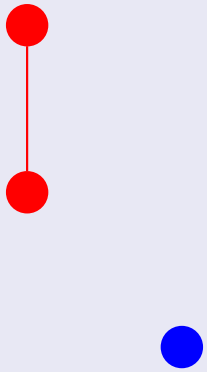
- Dos jugadores se turnan para remover aristas del árbol.
- En su turno cada jugador debe elegir exactamente una **arista** para remover.
- Cuando una arista es removida por un jugador también se elimina el subárbol que ya no queda conectado a la raíz del arbol.
- El jugador que no puede jugar pierde.



I. El calcentín mágico

Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:

- Dos jugadores se turnan para remover aristas del árbol.
- En su turno cada jugador debe elegir exactamente una **arista** para remover.
- Cuando una arista es removida por un jugador también se elimina el subárbol que ya no queda conectado a la raíz del arbol.
- El jugador que no puede jugar pierde.



I. El calcentín mágico

Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:

- Dos jugadores se turnan para remover aristas del árbol.
- En su turno cada jugador debe elegir exactamente una **arista** para remover.
- Cuando una arista es removida por un jugador también se elimina el subárbol que ya no queda conectado a la raíz del arbol.
- El jugador que no puede jugar pierde.



I. El calcentín mágico

Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:

- Dos jugadores se turnan para remover aristas del árbol.
- En su turno cada jugador debe elegir exactamente una **arista** para remover.
- Cuando una arista es removida por un jugador también se elimina el subárbol que ya no queda conectado a la raíz del arbol.
- El jugador que no puede jugar pierde.

Es un *juego imparcial*.

I. El calcentín mágico

Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:

- Dos jugadores se turnan para remover aristas del árbol.
- En su turno cada jugador debe elegir exactamente una **arista** para remover.
- Cuando una arista es removida por un jugador también se elimina el subárbol que ya no queda conectado a la raíz del arbol.
- El jugador que no puede jugar pierde.

Es un *juego imparcial*.

- los movimientos posibles dependen completamente del árbol.
- la única diferencia entre el jugador 1 y 2 es quien arranca el juego.

I. El calcentín mágico

Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:

- Dos jugadores se turnan para remover aristas del árbol.
- En su turno cada jugador debe elegir exactamente una **arista** para remover.
- Cuando una arista es removida por un jugador también se elimina el subárbol que ya no queda conectado a la raíz del arbol.
- El jugador que no puede jugar pierde.

Es un *juego imparcial*.

- los movimientos posibles dependen completamente del árbol.
- la única diferencia entre el jugador 1 y 2 es quien arranca el juego.

Algunos juegos que **no** son imparciales: ajedrez, poker.

I. El calcentín mágico

Problema Dado un árbol con raíz determinar si el primer o segundo jugador tiene estrategia ganadora en un juego con las siguientes reglas:

- Dos jugadores se turnan para remover aristas del árbol.
- En su turno cada jugador debe elegir exactamente una **arista** para remover.
- Cuando una arista es removida por un jugador también se elimina el subárbol que ya no queda conectado a la raíz del arbol.
- El jugador que no puede jugar pierde.

Es un *juego imparcial*.

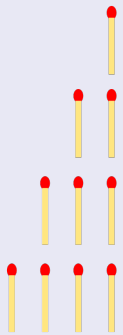
- los movimientos posibles dependen completamente del árbol.
- la única diferencia entre el jugador 1 y 2 es quien arranca el juego.

Algunos juegos que **no** son imparciales: ajedrez, poker.
Clásico juego imparcial: NIM

I. El calcentín mágico

Reglas del NIM

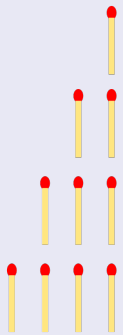
- *Se juega con pilas de fósforos.*



I. El calcentín mágico

Reglas del NIM

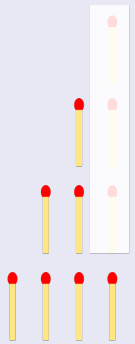
- *Se juega con pilas de fósforos.*
- *Dos jugadores se turnan para sacar fósforos de las pilas.*



I. El calcentín mágico

Reglas del NIM

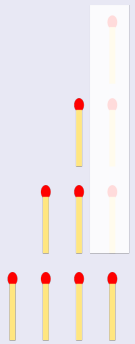
- *Se juega con pilas de fósforos.*
- *Dos jugadores se turnan para sacar fósforos de las pilas.*
- *En su turno cada jugador debe elegir una pila y sacar al menos un fósforo de esa pila.*



I. El calcentín mágico

Reglas del NIM

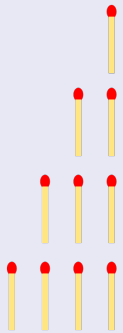
- *Se juega con pilas de fósforos.*
- *Dos jugadores se turnan para sacar fósforos de las pilas.*
- *En su turno cada jugador debe elegir una pila y sacar al menos un fósforo de esa pila.*
- *El jugador que no puede jugar pierde.*



I. El calcentín mágico

Valores del NIM

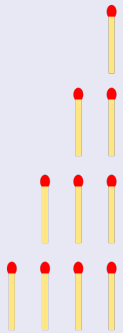
- *A cada posición del juego NIM se le puede asignar un* ***valor****.*



I. El calcentín mágico

Valores del NIM

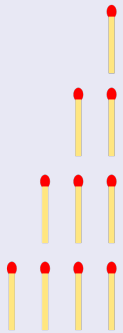
- *A cada posición del juego NIM se le puede asignar un **valor**.*
- *El valor representa las posibilidades de ganar para el siguiente jugador.*



I. El calcentín mágico

Valores del NIM

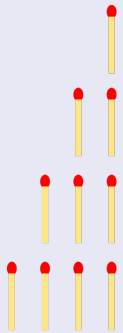
- *A cada posición del juego NIM se le puede asignar un **valor**.*
- *El valor representa las posibilidades de ganar para el siguiente jugador.*
- *Una posición con valor 0 significa una derrota para el siguiente jugador.*



I. El calcentín mágico

Valores del NIM

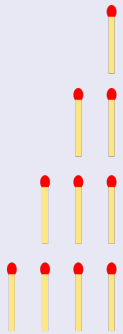
- *A cada posición del juego NIM se le puede asignar un **valor**.*
- *El valor representa las posibilidades de ganar para el siguiente jugador.*
- *Una posición con valor 0 significa una derrota para el siguiente jugador.*
- *XOR se utiliza para calcular los valores NIM.*



I. El calcentín mágico

Valores del NIM

- *A cada posición del juego NIM se le puede asignar un **valor**.*
- *El valor representa las posibilidades de ganar para el siguiente jugador.*
- *Una posición con valor 0 significa una derrota para el siguiente jugador.*
- *XOR se utiliza para calcular los valores NIM.*

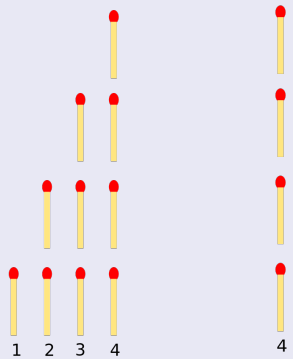


Jugar al NIM con N pilas de $a_1, a_2, \ldots, a_n$ fósforos es equivalente a jugar al NIM con una pila de $a_1 \oplus a_2 \oplus \ldots \oplus a_n$ fósforos.

I. El calcentín mágico

Valores del NIM

- *A cada posición del juego NIM se le puede asignar un **valor**.*
- *El valor representa las posibilidades de ganar para el siguiente jugador.*
- *Una posición con valor 0 significa una derrota para el siguiente jugador.*
- *XOR se utiliza para calcular los valores NIM.*



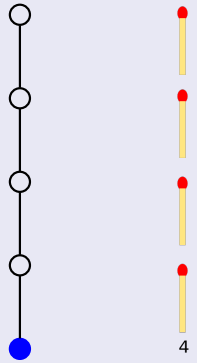
Jugar al NIM con N pilas de $a_1, a_2, \ldots, a_n$ fósforos es equivalente a jugar al NIM con una pila de $a_1 \oplus a_2 \oplus \ldots \oplus a_n$ fósforos.
Ejemplo: $(001)_2 \oplus (010)_2 \oplus (011)_2 \oplus (100)_2 = (100)_2$

I. El calcentín mágico

Cómo se relaciona NIM a nuestro juego de eliminar aristas en un árbol con raiz?

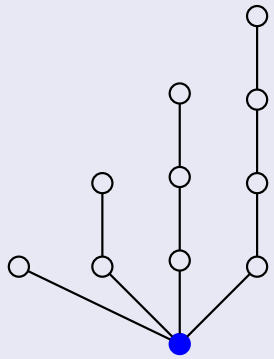
I. El calcentín mágico

Cómo se relaciona NIM a nuestro juego de eliminar aristas en un árbol con raiz? Si el árbol es un camino de longitud l nuestro juego es equivalente a jugar al NIM en una pila de l fósforos.



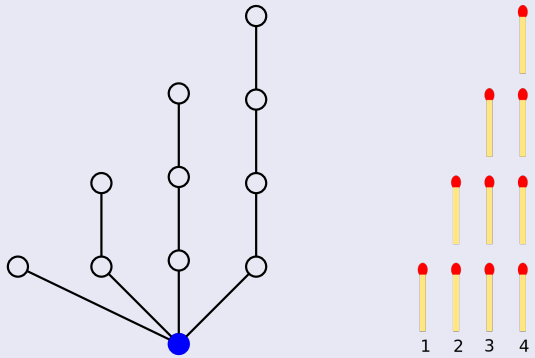
I. El calcentín mágico

Cómo se relaciona NIM a nuestro juego de eliminar aristas en un árbol con raiz?
Si el árbol es un camino de longitud l nuestro juego es equivalente a jugar al NIM en una pila de l fósforos.
Y si el árbol consiste de N caminos de longitud $l_1, l_2, \ldots, l_n$ que se conectamente solamente en la raiz?



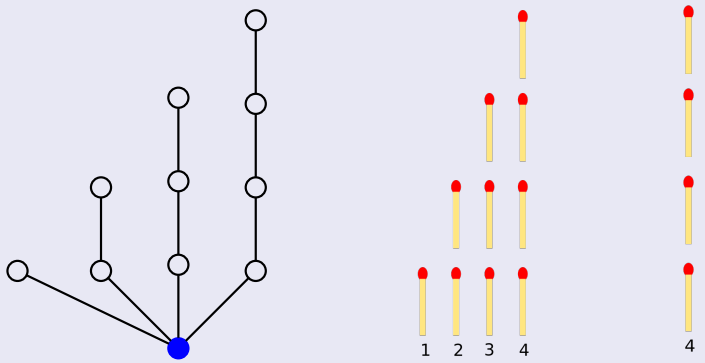
I. El calcentín mágico

Cómo se relaciona NIM a nuestro juego de eliminar aristas en un árbol con raiz?
Si el árbol es un camino de longitud l nuestro juego es equivalente a jugar al NIM en una pila de l fósforos.
Y si el árbol consiste de N caminos de longitud $l_1, l_2, \ldots, l_n$ que se conectamente solamente en la raiz? Es lo mismo que jugar al NIM con N pilas de $l_1, l_2, \ldots, l_n$ fósforos,



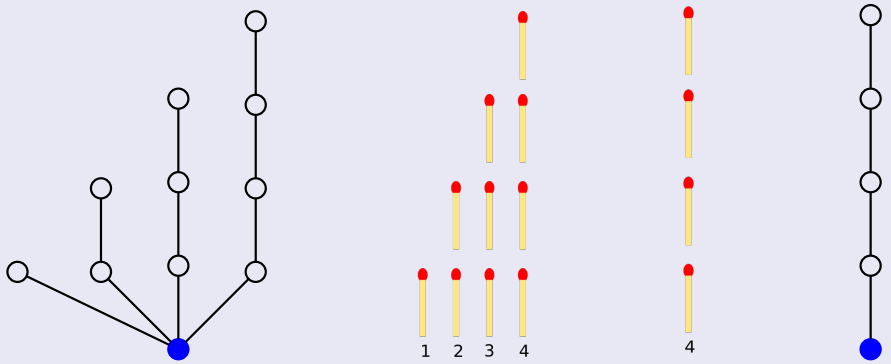
I. El calcentín mágico

Cómo se relaciona NIM a nuestro juego de eliminar aristas en un árbol con raiz? Si el árbol es un camino de longitud l nuestro juego es equivalente a jugar al NIM en una pila de l fósforos.
Y si el árbol consiste de N caminos de longitud $l_1, l_2, \ldots, l_n$ que se conectamente solamente en la raiz? Es lo mismo que jugar al NIM con N pilas de $l_1, l_2, \ldots, l_n$ fósforos, que es equivalente a jugar al NIM con un pila de $l_1 \oplus l_2 \oplus \ldots \oplus l_n$ fósforos.



I. El calcentín mágico

Cómo se relaciona NIM a nuestro juego de eliminar aristas en un árbol con raiz? Si el árbol es un camino de longitud l nuestro juego es equivalente a jugar al NIM en una pila de l fósforos.
Y si el árbol consiste de N caminos de longitud $l_1, l_2, \ldots, l_n$ que se conectamente solamente en la raiz? Es lo mismo que jugar al NIM con N pilas de $l_1, l_2, \ldots, l_n$ fósforos, que es equivalente a jugar al NIM con un pila de $l_1 \oplus l_2 \oplus \ldots \oplus l_n$ fósforos. Lo que a su vez, es equivalente a jugar nuestro juego en un camino de longitud $l_1 \oplus l_2 \oplus \ldots \oplus l_n$.

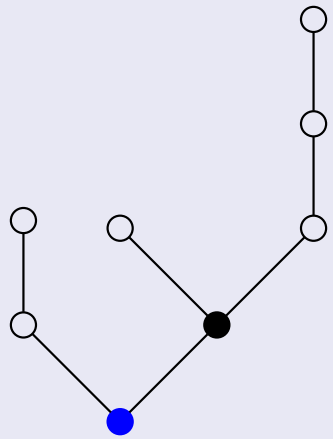


I. El calcentín mágico

Cómo se relaciona NIM a nuestro juego de eliminar aristas en un árbol con raiz?
Si el árbol es un camino de longitud l nuestro juego es equivalente a jugar al NIM en una pila de l fósforos.
Y si el árbol consiste de N caminos de longitud $l_1, l_2, \ldots, l_n$ que se conectamente solamente en la raiz?
Lo podemos reemplazar por un camino de longitud $l_1 \oplus l_2 \oplus \ldots \oplus l_n$.

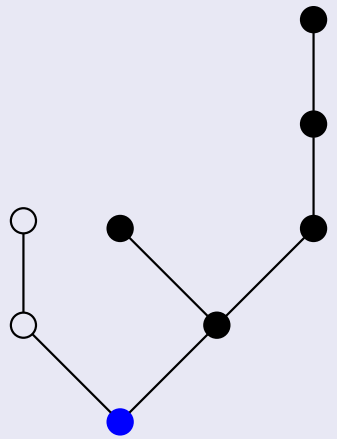
I. El calcentín mágico

Cómo se relaciona NIM a nuestro juego de eliminar aristas en un árbol con raiz?
Si el árbol es un camino de longitud l nuestro juego es equivalente a jugar al NIM en una pila de l fósforos.
Y si el árbol consiste de N caminos de longitud $l_1, l_2, \ldots, l_n$ que se conectamente solamente en la raiz?
Lo podemos reemplazar por un camino de longitud $l_1 \oplus l_2 \oplus \ldots \oplus l_n$.
Y si hay nodos intermedios con más de un hijo?



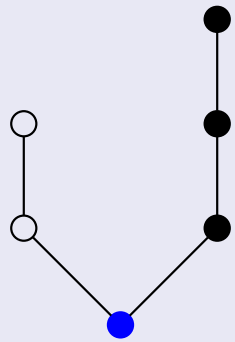
I. El calcentín mágico

Cómo se relaciona NIM a nuestro juego de eliminar aristas en un árbol con raiz?
Si el árbol es un camino de longitud l nuestro juego es equivalente a jugar al NIM en una pila de l fósforos.
Y si el árbol consiste de N caminos de longitud $l_1, l_2, \ldots, l_n$ que se conectamente solamente en la raiz?
Lo podemos reemplazar por un camino de longitud $l_1 \oplus l_2 \oplus \ldots \oplus l_n$.
Y si hay nodos intermedios con más de un hijo? Podemos reemplazar recursivamente cada subárbol por su camino equivalente hasta obtener un único camino.



I. El calcentín mágico

Cómo se relaciona NIM a nuestro juego de eliminar aristas en un árbol con raiz?
Si el árbol es un camino de longitud l nuestro juego es equivalente a jugar al NIM en una pila de l fósforos.
Y si el árbol consiste de N caminos de longitud $l_1, l_2, \ldots, l_n$ que se conectamente solamente en la raiz?
Lo podemos reemplazar por un camino de longitud $l_1 \oplus l_2 \oplus \ldots \oplus l_n$.
Y si hay nodos intermedios con más de un hijo? Podemos reemplazar recursivamente cada subárbol por su camino equivalente hasta obtener un único camino.



I. El calcentín mágico

Cómo se relaciona NIM a nuestro juego de eliminar aristas en un árbol con raiz?
Si el árbol es un camino de longitud l nuestro juego es equivalente a jugar al NIM en una pila de l fósforos.
Y si el árbol consiste de N caminos de longitud $l_1, l_2, \ldots, l_n$ que se conectamente solamente en la raiz?
Lo podemos reemplazar por un camino de longitud $l_1 \oplus l_2 \oplus \ldots \oplus l_n$.
Y si hay nodos intermedios con más de un hijo? Podemos reemplazar recursivamente cada subárbol por su camino equivalente hasta obtener un único camino.



I. El calcentín mágico

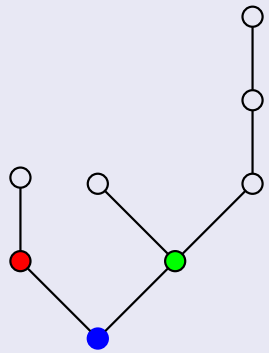
Cómo se relaciona NIM a nuestro juego de eliminar aristas en un árbol con raiz?
Si el árbol es un camino de longitud l nuestro juego es equivalente a jugar al NIM en una pila de l fósforos.
Y si el árbol consiste de N caminos de longitud $l_1, l_2, \ldots, l_n$ que se conectamente solamente en la raiz?
Lo podemos reemplazar por un camino de longitud $l_1 \oplus l_2 \oplus \ldots \oplus l_n$.
Y si hay nodos intermedios con más de un hijo? Podemos reemplazar recursivamente cada subárbol por su camino equivalente hasta obtener un único camino.
Si el camino tiene longitud 0, implica una derrota para el primer jugador. Caso contrario el primer jugador tiene estrategia ganadora.

I. El calcentín mágico

Tomando en cuenta lo anterior, podemos calcular el resultado de la siguiente manera:

```
int f(int v){
    int res = 0;
    for(auto u: G[v]) res ^= (1 + f(u));
    return res
}
```

Si $f(0) \neq 0$ el primer jugador tiene estrategia ganadora sino el segundo jugador la tiene.



$f(0) = (1+1) \oplus (1+2) = 1$