

Abrir paso a la línea del tren

Se está implementando un videojuego de lógica con la siguiente temática. Una compañía de construcción ferroviaria debe tender una línea entre varios países. En el camino se encuentran con varios obstáculos que deben moverse para que la línea del tren pueda pasar. Por ejemplo, la figura 1 muestra un conjunto de vehículos estacionados.

El jugador debe mover horizontalmente la menor cantidad de obstáculos (vehículos) de tal forma de que la línea del tren pueda pasar verticalmente por alguna de las columnas. Para mover un obstáculo, el jugador lo selecciona y decide si moverlo a la izquierda o a la derecha. Un obstáculo sólo puede moverse a un espacio vacío a la vez.

Mover un obstáculo tiene su costo. En el ejemplo dado son vehículos pequeños y moverlos un espacio cuesta una unidad de energía. Dado que en la historia del juego el problema es muy frecuente (hay muchos niveles), la compañía constructora contrata al jugador para que minimice la cantidad de energía que la compañía deba invertir y conseguir tender los rieles entre todos los países.

La solución no siempre es la columna que tiene más espacios vacíos. En el ejemplo de la figura 1, las columnas 2, 4 y 5 tienen dos espacios vacíos cada una, pero sólo una de ellas es la óptima. Para ayudar a encontrarla, la figura 2 muestra un modelo del ejemplo dado, donde un obstáculo se representa con un uno (1) y un espacio vacío con un cero (0).

Cada número en negrita en la parte inferior de cada columna, indica la cantidad de movimientos (unidades de energía) que se deben invertir para despejar esa columna. Por ejemplo, si el jugador quiere despejar la segunda columna (con índice 1) porque tiene dos espacios vacíos, se deben correr dos vehículos hacia la derecha en la primera fila (con índice 0) y dos vehículos en la tercera fila (con índice 2). En total 4 vehículos como indica el número en negrita. **Importante:** Note que para esta segunda columna sólo se pueden realizar movimientos hacia la derecha, pues no hay espacios vacíos hacia la izquierda. Su solución debe tener esta consideración en ambas direcciones.

Tras hacer este análisis para cada columna, se nota que la cuarta (con índice 3) requiere sólo dos movimientos, por tanto es la solución óptima. Si el jugador realiza esos dos movimientos, como se ve en la figura 3, la columna se despeja y los rieles podrán pasar en línea recta a través de ella.

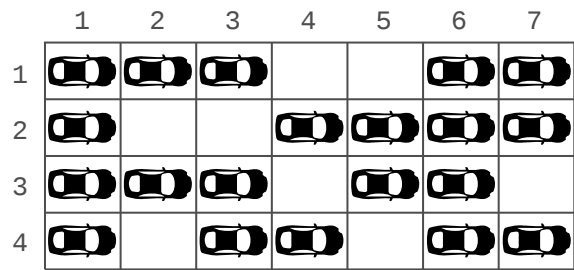


Figura 1. Un nivel del juego

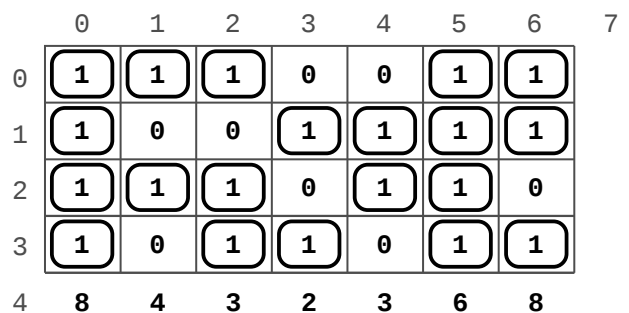


Figura 2. Modelo del nivel

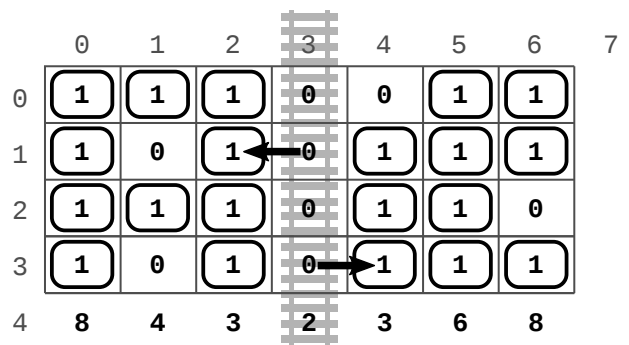


Figura 3. Modelo resuelto

Ejemplo de entrada:

```
4 7
1 1 1 0 0 1 1
1 0 0 1 1 1 1
1 1 1 0 1 1 0
1 0 1 1 0 1 1
```

Ejemplo de salida:

```
2 4
```

En el videojuego casual se quiere dar estrellas al jugador por su desempeño en el ahorro de energía. Se necesita un programa que dado un nivel del juego (una matriz con obstáculos), indique la cantidad mínima de energía para permitir que la línea del tren pase. El programa recibirá en la primera línea de la entrada estándar la cantidad de filas y columnas de la matriz. Después de una línea vacía, recibirá la matrix de obstáculos como muestra el siguiente ejemplo de entrada.

El programa deberá indicar en la salida estándar la cantidad mínima de movimientos (unidades de energía) para despejar la columna solución. Después de ese número se debe indicar la columna que resuelve el problema. Si varias columnas tienen esa cantidad mínima de energía se deben indicar separadas por espacios en blanco.

El videojuego usará esta salida para controlar el avance y premiar al jugador. En el ejemplo anterior la cantidad mínima de movimientos es 2. Se le dará al jugador $2 + 3 = 5$ unidades de energía en este nivel. Si el jugador despeja una columna mediante:

- 2 movimientos, obtendrá $5 - 2 = 3$ estrellas y avanza.
- 3 movimientos, obtendrá $5 - 3 = 2$ estrellas y avanza.
- 4 movimientos, obtendrá $5 - 4 = 1$ estrella y avanza.
- 5 movimientos, obtendrá $5 - 5 = 0$ estrellas pero avanza.

El jugador no podrá realizar un sexto movimiento porque se habrá quedado sin energía y tendrá que reiniciar el nivel. Las columnas solución podrían ser usadas por el videojuego para dar pistas al jugador.

Para que el nivel tenga solución, toda fila debe disponer de al menos un espacio vacío. Si ese no fuera el caso, el programa debe advertir a los desarrolladores imprimiendo `Invalid row N`, donde `N` debe reemplazarse por el número de línea que no tiene espacios libres.

Para su solución deberá implementar una clase completa en Java que resuelva el problema planteado. La solución debe apegarse a las buenas prácticas de programación. Es muy importante la separación en métodos. Al menos la referencia a la matriz debe implementarse como un atributo de la clase. Dado que el cálculo de los costos por cada columna es costo en tiempo de ejecución, guarde los resultados en un arreglo atributo de la clase. Por simplicidad, no es necesario documentar en JavaDoc la solución. Sin embargo, los comentarios de los algoritmos los puede escribir dentro del código o en una sección aparte en su cuaderno de examen. Por temática del examen debe usar estrictamente arreglos y matrices del lenguaje de programación Java y no clases de biblioteca para manejo de arreglos y matrices. Esto implica que **no está permitido** usar métodos de la clase `java.util.Arrays` o contenedores como `java.util.ArrayList`.