

Técnicas Avanzadas de Programación

Tarea #3. Algoritmos de Grafos

Fecha de Entrega: Jueves 19 de Octubre de 2017

Esta tarea consta de dos problemas que deberán ser resueltos por Ud.

1. Parte A

Pedro tiene dos robots controlados con control remoto, pero únicamente un control remoto. Suponga que los robots están en un tablero de $N \times N$ celdas y se desea llevarlos a las posiciones a y b del tablero. El control remoto únicamente permite girar a la izquierda o a la derecha 90° , o avanzar un paso hacia adelante, pero como hay un solo control remoto, cada vez que hagas un movimiento ambos robots lo harán. Por ejemplo, en la Figura 1, la imagen de la izquierda muestra la situación inicial. La flecha que apunta a la derecha es un robot de cara al este, y la flecha apuntando hacia arriba es un robot apuntando hacia el norte. Las posiciones a y b son los destinos de los robots. La imagen del centro muestra el resultado luego de mover un paso hacia adelante. La imagen de la derecha muestra el resultado luego de girar los robots a la izquierda.

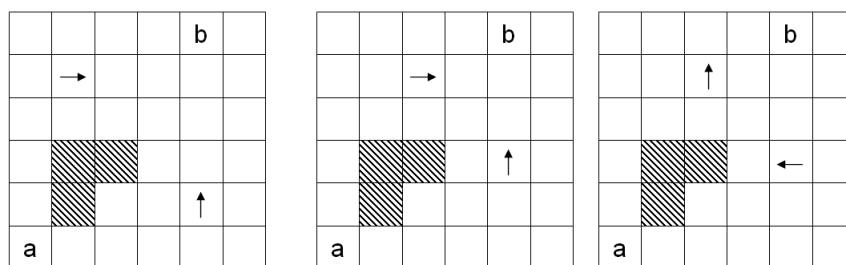


Figura 1: Movimiento de los robots. Las celdas grises son paredes

Ahora Pedro desea que lo ayudes a calcular el número mínimo de movimientos necesarios para llevar a los robots desde sus posiciones originales hasta sus destinos finales. Si a un robot se le ordena moverse hacia una pared el no lo hará (recuerde que son robots, no fantasmas). Si les ordena a los dos robots que

caminen a la misma posición tampoco lo harán (de nuevo, no son fantasmas). Refiérase a la Figura 2 para más información. En la imagen de la izquierda los robots pueden llegar a sus destinos en tres movimientos, aunque uno de los robots no se mueve en todos esos pasos. La imagen de la derecha muestra una situación en la que los robots no pueden avanzar ya que ocuparían la misma posición. Nótese que pueden haber paredes dentro del tablero. Los robots no pueden moverse dentro de una pared o fuera del tablero.

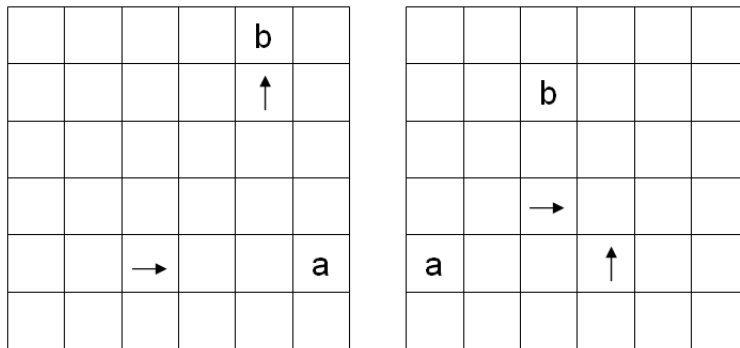


Figura 2: Algunas situaciones especiales

Entrada

La entrada debe ser leída por la entrada estándar (**stdin**). La entrada consiste de varios casos. La primera línea de cada caso comienza con un entero N que indica el tamaño del tablero ($N \times N$), $2 \leq N \leq 50$. Las siguientes N líneas describen el tablero, y contienen N caracteres cada una. Un '.' indica una celda vacía, un N, E, S o O indica la posición original del robot, de cara al Norte, Este, Sur u Oeste respectivamente; una D indica un destino para un robot (cualquiera de los robots), y '*' indica una pared.

La entrada culminará con un caso donde $N = 0$.

Ejemplo:

```
5
D...
N...S
....
*...*
....D
5
....
S..S.
....
```

```

.....
D..D.
3
SN.
***
.DD
0

```

Salida

La salida debe ser escrita por la salida estándar (`stdout`). Para cada caso de la entrada, imprima una línea con un entero indicando el mínimo número de movimientos necesarios para mover los robots a sus destinos. Imprima -1 si no hay forma de lograr que los robots lleguen a sus destinos.

Ejemplo para la entrada anterior:

```

8
3
-1

```

2. Parte B

Julio va a dar una charla y tiene preparadas láminas transparentes correctamente enumeradas. Lamentablemente las láminas no están en el orden correcto. Como las láminas son transparentes es posible ver los números que están dibujados en todas las láminas. En la Fig. 2 puede observarse un ejemplo de la situación.

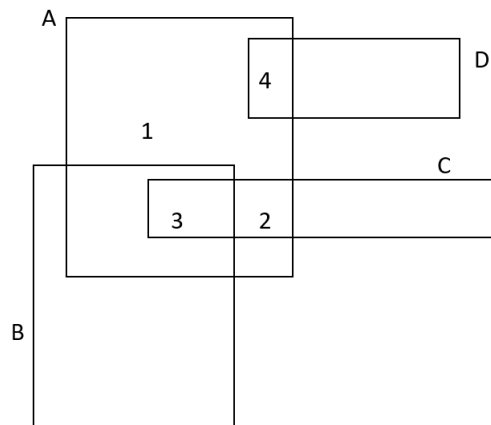


Figura 3: Las láminas sobre la mesa permiten ver todos los números

En este caso, las láminas originalmente estaban en el orden siguiente:

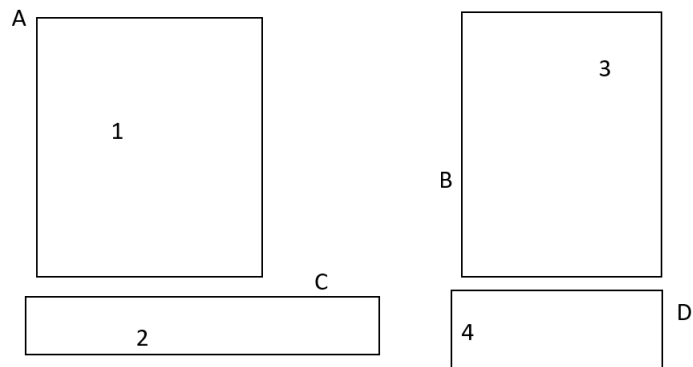


Figura 4: Láminas en el orden original

Ayuda a Julio a recordar el orden original de las láminas.

Entrada

La entrada comienza con el número de láminas N . ($1 \leq N \leq 100$). A continuación seguirán N líneas, cada una conteniendo la descripción de una lámina de la manera siguiente: $\text{Nombre}_i \ X1_i \ Y1_i \ X2_i \ Y2_i$. (Nombre_i es un string de hasta 10 letras mayúsculas). $X1_i \ Y1_i$ son las coordenadas de la esquina superior izquierda de la lámina i , $X1_i \ Y1_i$ son las coordenadas de la esquina inferior derecha de la lámina i . ($0 \leq X1_i < X2_i \leq 1000$), ($0 \leq Y1_i < Y2_i \leq 1000$).

A continuación seguirán N líneas, cada una indicando la posición de un número: $NX_i \ NY_i \ i$ ($1 \leq i \leq N$) ($0 \leq NX_i, NY_i \leq 1000$).

Ejemplo:

```
4
A 84 14 367 341
B 41 199 292 528
D 312 42 575 140
C 184 218 626 291
185 159 1
338 268 2
331 95 4
223 267 3
```

Salida

Imprima en una sola línea los nombres de las láminas en orden. En caso de existir más de un orden posible imprimir la palabra “Imposible”

Ejemplo:

A C B D

Esta tarea puede implementarse usando los lenguajes C, C++, Java o Python 3. Debe entregar el archivo fuente (un sólo archivo con extensión `cpp`, `c`, `py` o `java`). Esta tarea **puede hacerse en equipos de 2 personas**. La tarea debe subirse al link de dropbox enviado por correo. Para la parte A el archivo fuente debe llamarse “pa.cpp” (dependiendo de la extensión) y para la parte b “pb.cpp” (dependiendo de la extensión).

Al enviar el archivo debe colocar la cédula del primer integrantes en el campo de nombre de dropbox, y la cédula del segundo integrante en el campo de apellido. En caso de ser un sólo integrante, colocar la cédula en el campo de nombre, y en apellido repetir la cédula. Colocar la dirección de correo electrónico de cualquiera de los integrantes del proyecto.

GDTAP