

Tarea 3

Profesor: Diego Arroyuelo

Ayudantes: Tomás Berrios, Gabriel Carmona

`tomas.berrios@sansano.usm.cl`,

`gabriel.carmonat@sansano.usm.cl`

Fecha de Entrega: 11 de agosto, 2020

Plazo máximo de entrega: 5 horas.

Reglas del Juego

La presente tarea debe hacerse en grupos de 3 personas. Toda excepción a esta regla debe ser conversada con los ayudantes **ANTES** de comenzar la tarea. No se permiten de ninguna manera grupos de más de 3 personas. Pueden usarse los lenguajes de programación C, C++, Python, y Java. Esta tarea está pensada para una semana de trabajo, considerando que hay una semana intermedia de vacaciones.

Un Juego entre Hermanos

Dos hermanos juegan a un juego en el que disponen de un tablero de n^2 casilleros, de n filas por n columnas. Los casilleros son enumerados por filas: los casilleros de la primera fila tienen números $1, 2, \dots, n$, de izquierda a derecha; la segunda fila $n + 1, n + 2, \dots, 2n$; y así hasta la n -ésima fila, con números $n^2 - n + 1, n^2 - n + 2, \dots, n^2$. El juego consiste en moverse desde el casillero 1 hasta el n^2 , dando saltos entre los casilleros. Se permite saltar desde el casillero actual a cualquier otro casillero en donde el jugador no haya estado previamente.

Denotaremos con x_i al i -ésimo casillero visitado por el primer hermano en su secuencia de saltos. Entonces, si el primer hermano recorre p casilleros en total, estos se denotan x_1, x_2, \dots, x_p . Nótese que cada x_i en la secuencia es único, dada la restricción que impone el juego. Para el segundo hermano, los s casilleros recorridos se denotan con y_1, y_2, \dots, y_s .

Por ejemplo, dado un tablero de 3×3 , el primer hermano podría elegir recorrer los casilleros

$$1 \rightarrow 7 \rightarrow 5 \rightarrow 4 \rightarrow 8 \rightarrow 3 \rightarrow 9,$$

mientras el segundo

$$1 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 2 \rightarrow 8 \rightarrow 9.$$

Sin embargo su padre, quien los mira jugar, queda sorprendido de que los hermanos se mueven por separado. Si son hermanos, deberían moverse juntos. Ante este pedido, los hermanos deciden usar una estrategia para respetar lo más posible la secuencia de casilleros que pensaban recorrer por separado, y además respetar el pedido de su padre de hacer el recorrido juntos. A partir de sus recorridos, ellos quieren calcular el recorrido más largo que pueden hacer juntos, tal que los casilleros de ese recorrido aparezcan en ese orden en sus propios recorridos.

Para el ejemplo anterior, esto corresponde a la secuencia

$$1 \rightarrow 4 \rightarrow 8 \rightarrow 9,$$

de largo 4. Otra secuencia que maximiza los casilleros compartidos es

$$1 \rightarrow 5 \rightarrow 8 \rightarrow 9.$$

Su tarea es implementar una solución de programación dinámica tal que dadas las secuencias de casilleros de los dos hermanos, se conozca la longitud del recorrido más largo que pueden compartir. Se quiere resolver el problema de manera eficiente.

Hint: estudie el problema de la *subsecuencia común más larga* (longest common subsequence) entre dos strings.

Entrada de Datos

Los datos serán leídos desde la entrada standard, con el siguiente formato. La primera línea contiene un número entero positivo t , que indica la cantidad de casos de prueba. La siguiente línea contiene 3 enteros positivos n , p y s , indicando las dimensiones del tablero (n), la cantidad de casilleros que pensaba visitar el primer hermano (p), y la cantidad de casilleros que pensaba visitar el segundo hermano (s). Luego le siguen t pares de líneas, correspondientes a cada uno de los casos de prueba. La primera línea de dichos pares corresponde a los p casilleros del primer hermano, mientras que la segunda línea contiene los s casilleros del segundo hermano.

Un ejemplo de entrada es el siguiente:

```
1
4 8 7
1 10 3 4 13 11 5 16
1 5 9 10 4 13 16
```

Hint: para probar su programa de una mejor manera, ingrese los datos de entrada con el formato indicado en un archivo de texto (por ejemplo, el archivo `input.dat`). Luego, ejecute su programa desde la terminal, redirigiendo la entrada standard como a continuación:

```
./problema < input.dat
```

De esta manera evita tener que entrar los datos manualmente cada vez que prueba su programa, y evita errores.

Salida de Datos

La salida del programa debe hacerse a través de la salida standard, la cual contendrá t líneas (una línea por cada caso de prueba de la entrada). Cada línea contendrá el recorrido más largo que pueden compartir los hermanos para el caso de prueba correspondiente. Dicho recorrido debe mostrarse en una única línea de la salida, con cada valor separado por un único espacio.

La salida correspondiente a los datos de entrada mostrados anteriormente es la siguiente:

```
1 10 4 13 16
```

Entrega de la Tarea

La entrega de la tarea debe realizarse enviando un archivo comprimido llamado

`tarea2-apellido1-apellido2-apellido3.tar.gz`

(reemplazando sus apellidos según corresponda), o alternatively usando formato zip, en el sitio Aula USM del curso, a más tardar el día 11 de agosto, 2020, a las 23:59:00 hrs (Chile Continental), el cual contenga:

- Los archivos con el código fuente necesarios para el funcionamiento de la tarea.
- `NOMBRES.txt`, Nombre y ROL de cada integrante del grupo.
- `README.txt`, Instrucciones de compilación en caso de ser necesarias.
- `Makefile`, Instrucciones para compilación automática.