



Laboratorio

Curso: Introducción a la Programación.

Docente: Alejandro Alfaro Quesada. (aalfaroq@utn.ac.cr)

Fecha de entrega: domingo 20 de julio del 2025.

Hora de entrega: 11:59 pm.

Indicaciones Generales.

- El sistema debe ser programado en el lenguaje Python.
- La prueba se desarrolla bajo la modalidad de laboratorio, con tiempo de inicio y fin predeterminado.
- La prueba es estrictamente individual.
- Toda prueba evaluada es considerada como un único proyecto, es decir, aunque la evaluación sea dividida por secciones, la completitud de uno o más rubros puede afectar la funcionalidad del sistema, y esto, podría influir en la nota final.
- Si algún estudiante es sorprendido en actos fraudulentos, ya sea que los provoque o los consienta, su prueba y la de los demás implicados, será automáticamente anulada.
- El docente ejecuta el código fuente que está en el proyecto enviado para comprobar que realiza lo solicitado. Por lo que, el docente NO modificará ninguna línea de código, por lo tanto, debe asegurarse que el código fuente funciona correctamente.
- El proyecto con el código solo se recibe y se valida desde el Campus Virtual de la UTN. Por cualquier otro medio que se envíe al docente no será revisado.
- Para la explicación del trabajo el estudiante debe estar de manera presencial en la clase, de lo contrario, se rebajan 15 puntos de la nota que obtenga en el momento de la evaluación.

Red de Caminos de Energía

Nota: Para resolver este ejercicio se prohíbe el uso de librerías como numpy, pandas o similares de alto nivel (si considera alguna y tiene dudas, consulte al docente), debe ser utilizando únicamente las estructuras de datos vistas en clase; además, se prohíbe el uso de archivos de cualquier formato; el uso de las librerías o archivos invalida por completo la prueba.

El sistema debe ser capaz de encontrar el camino más largo de lado izquierdo al lado derecho consumiendo la menor cantidad de energía en una Red de Caminos de Energía; para esto se tendrá una serie de condiciones que se deben ir cumpliendo. No intente resolver las posibles soluciones porque pueden ser muchas.

Vamos a considerar una matriz de $N \times N$ (puede ser de cualquier tamaño que vayan desde 3×3 hasta 50×50) y que cada celda de la matriz esté representada por una unidad de energía (valor numérico entero positivo o negativo). Debe solicitar al inicio la dimensión de la matriz en cuanto a la cantidad de filas y columnas que tendrá.

La matriz se llenará con números aleatorios comprendidos entre el 1 y el 15 (ambos números se incluyen también); en donde el 75% de los números serán positivos y el 25% serán números negativos. La primera mitad de las columnas de la matriz estará compuesta únicamente por números positivos, la otra mitad tendrá tanto números positivos como negativos. Debe imprimir la matriz que se genera.

Para poder ir validando las celdas y encontrar el camino más largo entre la matriz, se debe cumplir con lo siguiente:

- Cuando el sistema inicia, el nivel de energía con la que cuenta es de 100, pero eso se puede ir agotando si se toman malas decisiones en el momento de buscar el camino. No se puede gastar más del nivel disponible.
- El algoritmo siempre iniciará su recorrido desde la columna cero sin importar la fila.
- Desde cualquier celda, es posible moverse hacia arriba, abajo, izquierda o derecha, pero sólo si el valor de la celda destino es estrictamente mayor que el valor actual. Cada movimiento cuesta 1 punto de energía, que se estaría restando del total.
- El sistema permite movimientos diagonales, igual cumpliendo que sólo si el valor de la celda destino es estrictamente mayor que el valor actual; y para este caso, cada movimiento diagonal cuesta 2 puntos de energía, que se estarían restando del total. El algoritmo debe ser capaz de determinar si vale la pena o no moverse en diagonal.

- Considere que con las dos restricciones anteriores no es solamente encontrar el camino más largo, sino que también consumiendo la menor cantidad de nivel de energía o que no se gaste.
- Los números enteros negativos que se generaron al inicio son para representar energía negativa; es decir, si se cae en alguna celda con un valor negativo, se debe restar 3 puntos de energía al total que se tiene en ese momento. Por lo tanto, el algoritmo debe ser capaz de poder evitar al máximo pasar por una celda de éstas. No quiere decir que debe restar valores entre celdas, solo afecta el nivel de energía.
- También, cada vez que se logre avanzar, el algoritmo debe validar que puede moverse en caminos que formen números primos; es decir, cada vez que se suma el valor de la celda actual y la nueva celda, el resultado debe ser un número primo.

En resumen:

1. Se debe mover a celdas de mayor valor.
2. Movimientos normales (arriba/abajo/izquierda/derecha) cuestan 1 punto de energía.
3. Movimientos diagonales cuestan 2 puntos de energía.
4. Algunas celdas tienen energía negativa que restan 3 puntos de energía.
5. La suma del valor actual más el valor destino debe ser número primo.
6. Energía inicial limitada a un nivel de 100.

Por lo tanto, el ejercicio tiene tres objetivos fundamentales:

1. Encontrar el camino más largo de izquierda a derecha.
2. El máximo de energía que se puede acumular con ese camino.
3. El menor número de movimientos que permitan minimizar el riesgo de perder energía en el camino.

Veamos un posible ejemplo:

Considerando una matriz de 5x5, con números enteros positivos y negativos generados aleatoriamente, sería algo así:

f/c	0	1	2	3	4
0	12	14	-3	18	22
1	11	17	4	16	28
2	-5	7	10	-8	20
3	6	15	13	19	-9
4	2	5	23	24	21

Caso #1: Si el algoritmo decide iniciar en la posición (0,0) cuyo valor es el 12 tiene las siguientes opciones:

1. Si se mueve a la derecha (0,1) sería tomar el valor 14 el cual es válido porque es mayor.
2. Al sumar el 12 más el 14, el resultado sería 26 lo cual no es un número primo, entonces se descarta esta alternativa.
3. Entonces si se mueve hacia abajo a la posición (1,0), el valor que se encuentra es el 11, pese a que 12 más 11 da 23 y sería un número primo, no se cumple la condición de que el número siguiente debe ser mayor, porque 11 no es mayor que 12.
4. Como conclusión el algoritmo debe determinar que iniciar desde la posición (0,0) para este caso no es factible.

Caso #2: Si el algoritmo decide iniciar en la posición (1,0) cuyo valor es el 11 tiene las siguientes opciones:

1. Si se mueve a la derecha (1,1) sería tomar el valor 17 el cual es válido porque es mayor.
2. Al sumar el 11 más el 17, el resultado sería 28 lo cual no es un número primo, entonces se descarta esta alternativa.
3. Entonces si se mueve hacia abajo a la posición (2,0), el valor que se encuentra es el 5 (recuerde que, aunque sea negativo no es que se resta al valor de la otra posición si no que resta al nivel de la energía), al sumar 11 más el 5 el resultado da 16 lo cual no es un número primo, entonces se descarta esta alternativa.
4. Si se mueve hacia arriba a la posición (0,0), el valor que se encuentra es el 12, al sumar 11 más el 12 el resultado sería 23 lo cual sí es un número primo además de que se cumple la condición que el siguiente número es mayor ($11 < 12$) lo cual sería un movimiento permitido.
5. Como conclusión quiere decir que iniciando desde (1,0) y moviéndose hacia arriba a la posición (0,0) podría ser un buen inicio de camino, aunque, el algoritmo podría determinar que cuando continúe por el camino del (0,0) se dé cuenta que no es la mejor alternativa, por lo ya explicado (y analizado en su momento por el algoritmo) en el caso #1.

Como resultado final, el sistema debe imprimir el siguiente reporte:

1. La matriz original.
2. Cada uno de los pasos que fue dando con el análisis realizado de lo que fue sucediendo.
3. La cantidad de energía consumida (de acuerdo con los movimientos).
4. La cantidad del nivel de energía que sobró.
5. El camino más largo representado por todas las posiciones (X,Y) por donde pasó.



Documentación.

Debe crear y subir junto con la solución un documento (Word) en donde explique lo siguiente:

Lógica del algoritmo: debe explicar claramente cómo funciona el algoritmo que utilizó para resolver el problema, incluyendo cómo se validó el paso entre celdas para encontrar el camino elegido, validaciones de restricciones como números primos, uso de energía y penalización por negativos; en cada explicación coloque una imagen del código donde realiza el proceso, así como datos que ud considere relevantes para explicar lo que desarrolló.