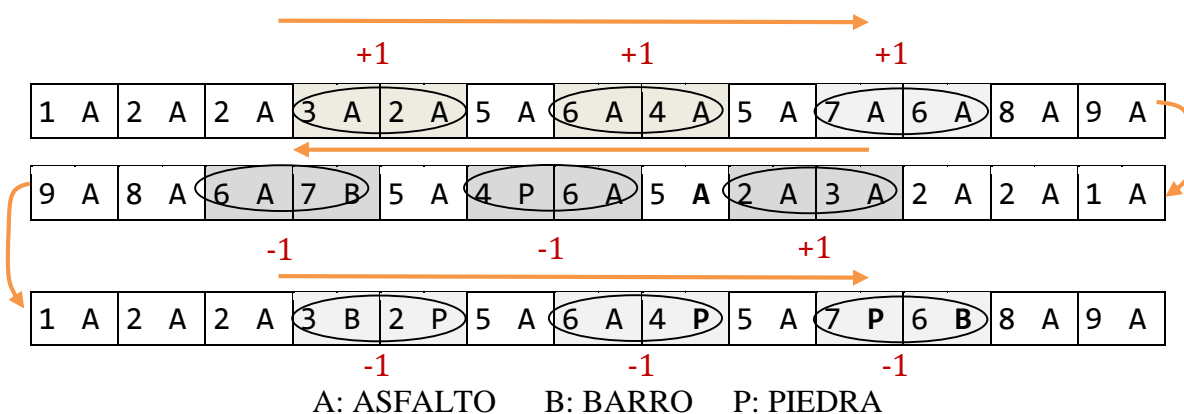


Problema 1: Mapa de juego**Enunciado:**

Un mapa de un juego está modelado a través de una matriz de casillas, y cada casilla posee dos propiedades: tipo de casilla, e inclinación. Tipo de casilla es un tipo enumerado que puede ser ASFALTO, PIEDRA o BARRO, mientras que la inclinación es un valor entero positivo que describe la altura de dicha casilla. En dicha matriz de casillas, se desea realizar un recorrido. En dicho recorrido, el hecho de pasar de una casilla a otra, puede tener diferentes recompensas ó penalizaciones. Para ser más concretos, si la transición es de una casilla a otra **del mismo tipo** y las casillas tienen una inclinación positiva (es decir, la inclinación de la casilla origen es mayor que la casilla destino) entonces tendremos una recompensa de +1 en experiencia. Por otro lado, si la transición es entre casillas diferentes y la inclinación es positiva tendremos una penalización de -1 en experiencia. En cualquier otro caso, el hecho de pasar de una casilla a otra no tendrá ni recompensa ni penalización. Se quiere implementar la función *calculaRecompensas*, que dada una matriz de casillas como las descritas anteriormente permita calcular el total de recompensas/penalizaciones que se obtendría de recorrer dicha matriz en zig-zag, es decir, en primer lugar la primera fila de izquierda a derecha, tras esto la segunda fila de derecha a izquierda, tras esto la tercera fila de izquierda a derecha, etc. Veamos un ejemplo de matriz de casillas, es decir, la entrada del problema:

1 A	2 A	2 A	3 A	2 A	5 A	6 A	4 A	5 A	7 A	6 A	8 A	9 A
9 A	8 A	6 A	7 B	5 A	4 P	6 A	5 A	2 A	3 A	2 A	2 A	1 A
1 A	2 A	2 A	3 B	2 P	5 A	6 A	4 P	5 A	7 P	6 B	8 A	9 A

Veamos un ejemplo donde la función *calculaRecompensas* es un resultado de -1.



No se tendrá en cuenta el salto de una fila a otra en el cálculo de penalizaciones o recompensas, es decir, el salto de la casilla 13 de la primera fila a la primera casilla de la segunda fila, no produce ni recompensa ni penalización. Cada casilla tiene una inclinación y una sigla correspondiente al tipo de celda. En este caso las casillas 4 y 5 de la primera fila tienen una inclinación positiva pero son del mismo tipo por lo tanto producen una recompensa positiva, sin embargo las casillas 7 y 8 de la segunda fila tienen una inclinación positiva pero son de tipos diferentes con lo que produce una penalización.

Preguntas

1. Defina la función recursiva no final *calculaRecompensas*.
2. Implemente una función **recursiva no final** empleado el lenguaje de programación Java.
3. Implemente una función **recursiva final** a partir de la definición anterior.
4. A partir de las funciones anteriores, diseñar e implementar un **algoritmo iterativo** siguiendo las plantillas vistas en clases.
5. Indique el tamaño del problema, $T(n)$ y la complejidad del algoritmo.