Trabajo Práctico 3

"Y dale con los mapas..."

Introducción a la Computación

1^{er} cuatrimestre 2019

Observaciones generales:

- El trabajo se debe realizar en grupos de tres personas.
- El código fuente debe estar bien comentado. Cualquier aclaración adicional que se considere necesaria debe ser incluida como comentarios.
- El código fuente debe enviarse por mail a la lista de docentes de la materia: `icm-doc@dc.uba.ar`, indicando los integrantes del grupo.
- El programa entregado debe compilar correctamente usando Python 3.6.7, que está instalado en las computadoras del laboratorio.
- Se evaluará la correctitud, claridad y modularidad del código entregado.
- La fecha límite de entrega es el jueves 13 de junio a las 23:59.

En este trabajo práctico vamos a reimplementar las mismas funcionalidades del TP2, ahora encapsuladas en un TAD Mapa, que además proveerá algunas funcionalidades adicionales.

Se pide implementar el tipo Mapa que tenga definidas (al menos) las siguientes operaciones, con el orden de complejidad algorítmica indicado en cada caso:

TAD MAPA

- *Mapa(nombre_archivo)* $\rightarrow$ Mapa: Construye un nuevo mapa, leyéndolo de un archivo de texto, tal como se especificó en el TP2, en **O(*longitud del archivo*)**.
- *m.es_posición_válida(pos)*: Dice si la posición pos está definida en el mapa m, en **O(1)**.
- *m.posición(pos)*: Devuelve el valor de la posición pos del mapa m, en **O(1)**.
  Precondición: *m.es_posición_válida(pos)*
- *m.alto()*: Devuelve el alto del mapa m, en **O(1)**.
- *m.ancho()*: Devuelve el ancho del mapa m, en **O(1)**.
- *m.cantidad_paredes()*: Devuelve la cantidad de paredes del mapa m, en **O(1)**.
- *m.corredor_horizontal_más_largo()*: Devuelve la longitud del corredor horizontal más largo del mapa m, según lo definido en el TP2, en **O(*ancho* $\times$ *alto*)**.
- *m.densidad_arquitectónica()*: Devuelve la densidad arquitectónica del mapa m, según lo definido en el TP2, en **O(1)**.

- *m.alcanzar_con_mano_derecha(pos_inicial, pos_destino)*: Devuelve True si y solo si un explorador dentro de la grilla del mapa podría partir desde la posición inicial y llegar a la posición destino usando el método de la mano derecha. Este método para recorrer ciertos tipos de laberintos consiste en avanzar siempre manteniéndose con la mano derecha tocando la pared; si la pared continúa para adelante, uno avanza; si en un momento tal pared se vuelve una esquina que gira hacia la derecha, uno debe seguirla; etc. Asumimos que el explorador recorre un espacio vacío por paso (respecto a la grilla: una posición para arriba, para abajo, para la izquierda, o para la derecha), y que no puede moverse en diagonal. La función debe devolver esta respuesta en **O(*ancho* $\times$ *alto*)**.
  Precondición: *m.posición(pos_inicial)* $= 0$ (i.e. el explorador empieza en un espacio vacío). El mapa m no tiene bloques de espacios vacíos de 2x2 o mayores.

De ser necesario, se pueden definir TADs auxiliares. La entrega del trabajo práctico debe incluir:

1. un informe que contenga:
   a) las estructuras de representación propuestas para los TADs que se hayan utilizado;
   b) los invariantes de representación de las estructuras propuestas (en español);
   c) los algoritmos en pseudocódigo, justificando en cada caso que cumplen con los órdenes requeridos;
2. la implementación del TAD Mapa en Python.

El TAD Mapa debe implementarse en una clase llamada `Mapa`. Su código fuente debe estar en un archivo llamado `mapa.py` y respetar la signatura de las siguientes operaciones (entre otras):

```
class Mapa:
  def __init__(self, nombre_archivo)
  def es_posicion_valida(self, pos)
  def posicion(self, pos)
  def alto(self)
  def ancho(self)
  def cantidad_paredes(self)
  def corredor_horizontal_mas_largo(self)
  def densidad_arquitectonica(self)
  def alcanzar_con_mano_derecha(self, pos_inicial, pos_destino)
```

Observaciones:

- Definir todas las funciones auxiliares que se consideren necesarias.
- No está permitida la importación de ninguna biblioteca, con la excepción de `sys` para el manejo de los argumentos del programa principal con `sys.argv`.