



IIC2115 – Programación como Herramienta para la Ingeniería (II/2019)

Actividad de participación 2 - Estructuras de datos

Objetivos

- Consolidar los conocimientos de estructuras de datos.
- Comprender el uso estructuras de datos y sus aplicaciones en redes.

Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **P02**.
- **Entrega:** jueves 5 de septiembre de 2019
- **Tiempo:** 80 minutos desde la publicación del enunciado en Syllabus
- **Formato de entrega:** un archivo jupyter notebook llamado (**solucionP02.ipynb**) y un archivo Python (**solucionP02.py**). ambos archivos deben contener la misma versión de la actividad y deben estar ubicados en la carpeta **P02**. No se debe subir ningún otro archivo a la carpeta. Utilice múltiples celdas de texto y código para facilitar la revisión de su trabajo.
- **Descuentos:** Las actividades que no cumplan el formato tendrán descuento de 0.5 pts.
- **Atrasos:** La no recepción del archivo a tiempo implica nota **1.0** sin excepción.
- **Tareas con errores de sintaxis y/o que generen excepciones serán calificadas con nota 1.0.**

Ejercicio

¡El Tigre ha sido herido! El famoso héroe de Progra Muy Muy Lejana, se encuentra débil en algún punto del bosque encantado. Sus fieles escuderos Pablo y Felipe (P & F) deciden acudir en su ayuda. Para ello, cuentan con el mapa **bosque.txt** en donde cada letra representa una casilla del bosque. El formato del mapa se describe a continuación:

- O : representa una casilla vacía. Desde cada casilla vacía se puede ir a otra (también vacía) que esté inmediatamente arriba, abajo, a la izquierda o a la derecha, si es que estas existen.
- X : representa una casilla ocupada, es decir, no se puede transitar por ella.

El sistema de coordenadas del mapa se representará de la siguiente forma:

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & \dots & a_{0m} \\ a_{10} & a_{11} & a_{12} & \dots & a_{1m} \\ a_{20} & a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n0} & a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}$$

Para poder rescatar a El Tigre, P & F te contratan como experto en Estructuras de Datos. Pero antes es necesario familiares con estructuras de datos un poco más complejas. Para eso vamos a trabajar en conjunto creando la estructura necesaria para rescatar al Tigre.

- Vamos a crear una función **leer_mapa(path)** que lea un mapa **.txt** con el formato descrito anteriormente y que reciba como parámetro el **path** (ruta al archivo). Debe retornar una lista de las filas del mapa, en donde cada fila es un string con una secuencia de X y O . Recuerde que si el archivo **.txt** se encuentra en la misma carpeta que el **.ipynb**, basta con que el path sea el nombre del archivo **.txt**.
- Ahora vamos a crear la función **crear_grafo(lista)** que reciba como input la lista anterior y cree un grafo en donde cada nodo sea una casilla vacía. Cada nodo debe guardar información sobre su ubicación en el mapa y tener una lista de los nodos adyacentes. Se espera que el output de esta función sea un diccionario de la forma $\{(\mathbf{i}, \mathbf{j}): \mathbf{nodo}\} (\forall (i, j) : a_{ij} = O)$ en donde cada llave sea una coordenada de la forma (\mathbf{i}, \mathbf{j}) explicada anteriormente. Los valores del diccionario deben ser objetos de la clase **Nodo**, en donde cada uno guarde en una lista sus vecinos en el orden: (arriba, izquierda, abajo, derecha). Se adjunta una visualización del orden de prioridades.

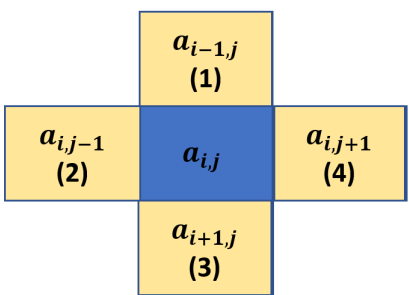


Figure 1: Orden de prioridades

Tareas a realizar

1. Cree la función **buscar_tigre(origen, destino)** que modele el movimiento de P & F por el bosque. P & F siempre que puedan se moverán en el orden descrito en la figura anterior. Para no repetir casillas, P & F marcarán los lugares por donde pasan. Cuando no tienen opciones disponibles ¹, P & F se devuelven a la última casilla que marcaron y vuelven a repetir el proceso. Se espera que la función retorne la secuencia de nodos recorridos por P & F hasta encontrar a El Tigre y la cantidad de casillas marcadas (lista, entero). **Nota:** origen y destino son tuplas con coordenadas en el formato (i, j) descrito anteriormente.
2. ¡Los malvados Hans y Rajaime (H & R) han descubierto que su enemigo El Tigre está débil y han decidido acabar con él! H & R deciden exterminar cada casilla vacía para asegurarse de que su rival no sobreviva. Para ello exterminan cada casilla en el mismo orden que P & F (arriba, izquierda, abajo, derecha). La diferencia está en que luego visitarán cada casilla exterminada (en el orden el que las exterminaron) y repetirán el mismo proceso. Continuarán así hasta que encuentren a El Tigre. P & F te piden crear la función **exterminar_tigre(origen, destino)**, que reciba un origen y un destino, y modele el recorrido hecho por H & R hasta exterminar a El Tigre. Se espera que la función retorne la secuencia de nodos recorridos por H & R hasta encontrar a El Tigre y la cantidad de casillas exterminadas (lista, entero). **Nota:** origen y destino son tuplas con coordenadas con el formato (i,j) descrito anteriormente.
3. Has descubierto tres posibles ubicaciones de El Tigre. P & F comenzarán a moverse al mismo tiempo que H & R desde puntos diferentes del bosque. Se te pide crear la función **se_salva_tigre**, que ejecute **buscar_tigre** y **exterminar_tigre** para cada ubicación y determinar, de acuerdo a la cantidad de casillas marcadas, si El Tigre será rescatado o exterminado. Debe utilizar el archivo **bosque.txt**. Asuma que el tiempo que toma visitar (P & F) y exterminar (H & R) una casilla es el mismo. Retorne True o False si El Tigre se salva o no (en caso de igualdad entre la cantida de casillas, asuma que es exterminado). Los puntos a reportar son:
 - (a) Origen P & F: (7, 8). Origen H & R: (6, 10). Destino: (7, 7)
 - (b) Origen P & F: (17, 12). Origen H & R: (6, 12). Destino: (2, 12)
 - (c) Origen P & F: (7, 5). Origen H & R: (8, 12). Destino: (7, 7)
4. **Proponer posición (Bonus):** ¡Has podido llamar a El Tigre! Para salvarlo, debes recomendarle un lugar donde esconderse, es decir, un punto del tablero en donde P & F lleguen en un menor

¹Si no existe ninguna casilla adyacente (arriba, izquierda, abajo, derecha) vacía y sin marcar.

número de pasos que H & R. Asuma que ambos comenzarán a buscarlo desde el punto (12, 12). Debe utilizar el archivo **bosque.txt**. Justifica tu respuesta ejecutando **buscar_tigre** y **exterminar_tigre** y comparando la cantidad de casillas marcadas. **Nota:** debe ser un punto diferente a los del punto anterior.

Importante: puedes asumir que El Tigre siempre estará en una casilla vacía alcanzable, es decir, nunca ocurrirá el caso en donde **buscar_tigre** o **exterminar_tigre** recorran todo el mapa sin encontrarlo.

Ejemplo

Para entender cómo funcionan los dos algoritmos, se adjunta una representación del mapa de prueba **test.txt** y cuatro escenarios diferentes. La simbología es la siguiente:

- Casillas vacías representadas por cuadros azules
- Casillas ocupadas representadas por cuadros negros
- Origen representado en verde
- Destino representado en naranja
- Casillas numeradas en amarillo representan camino recorrido por P & F
- Casillas numeradas en morado representan exterminio hecho por H & R

Escenario 1: P & F encuentran al Tigre eficientemente

El caso más simple para P & F es que El Tigre esté directamente arriba de ellos. Si no hay obstáculos, lo encontrarán muy rápidamente.

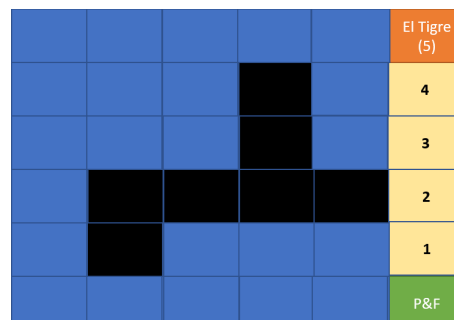


Figure 2: Escenario 1

Escenario 2: P & F encuentran a El Tigre de forma poco eficiente

Este escenario evidencia lo poco eficiente que puede ser la búsqueda de El Tigre para algunos escenarios en donde se esperaría encontrarlo rápidamente.

P&F	19	18	17	16	15
1	El Tigre (20)				14
2					13
3					12
4		8	9	10	11
5	6	7			

Figure 3: Escenario 2

Escenario 3: H & R exterminan a El Tigre eficientemente

Si El Tigre está cerca, H & R lo exterminarán rápidamente, da igual en qué dirección se esconda.

H&R	2				
1	El Tigre (4)				
3					

Figure 4: Escenario 3

Escenario 4: H & R encuentran a El Tigre de forma poco eficiente

Para algunas configuraciones simples para P & F, H & R demoran bastante en eliminar a El Tigre debido a su política.

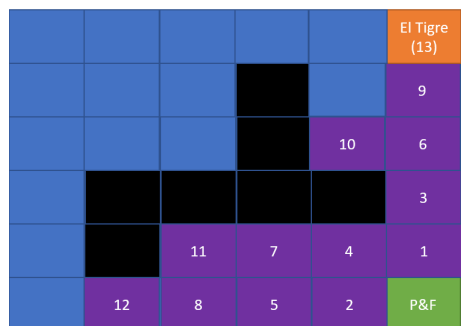


Figure 5: Escenario 4

Política de Integridad Académica

“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por “copia” o “plagio” se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.