



IIC2115 – Programación como Herramienta para la Ingeniería (I/2018)

## Tarea 2 - Programación Funcional, Estructuras de Datos y Algoritmos

### Objetivos

- Implementar el algoritmo de navegación A\*.
- Usando estructuras de datos, construir un mapa a partir de un archivo.
- Obtener métricas de desempeño mediante el uso de programación funcional.
- Redactar un informe usando L<sup>A</sup>T<sub>E</sub>X.

### Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** GitHub
- **Fecha:** 11 de mayo
- **Hora:** 23:59
- **Desarrollo en parejas o individual**
- Las inscripciones para las parejas son hasta el viernes 27 de Abril a las 23:59 y solo se aceptan las inscritas por este FORMULARIO.

# Introducción

Usted ha sido contratado por Edison, la conocida marca de autos eléctricos, para ayudar a desarrollar su línea de autos autónomos, a estrenar el año próximo.

Para este efecto, su *team manager* le ha encomendado la siguiente tarea: desarrollar un sistema de navegación que trabaje sobre un mapa dado, en donde se implementen algoritmos de búsqueda recursivos (NO iterativos). Sin embargo, al enviar los mapas hubo un corte de luz, por lo que los archivos están corruptos. Debido a esto, deberá pre-procesarlos para lograr obtener los datos reales del mapa. Una vez procesados, deberá implementar una estructura de datos que represente dicho mapa, para aplicar la navegación de su auto sobre él, y recopilar métricas de desempeño. Para esto, su sistema deberá poder realizar consultas valiéndose de técnicas de programación funcional. Su reporte/tarea (escrito en L<sup>A</sup>T<sub>E</sub>X) informará acerca de las conclusiones relevantes del caso.

## Preprocesamiento del archivo

A usted se le entregará un archivo *mapa.txt* que contiene un mapa defectuoso. Su misión es usar programación funcional para descifrar los contenidos, y así poder trabajar con estos. La estructura el archivo es la siguiente:

- En cada espacio del mapa solo cabe 1 vehículo.
- Un obstáculo se mostrará como una “X” en el mapa. Un vehículo no puede estar sobre un obstáculo.
- Cada espacio que no sea un obstáculo puede ser utilizado por el vehículo para movilizarse.
- Cada espacio tendrá un número que indicará la dificultad que tiene el vehículo para transitar por él.
- Un vehículo puede moverse entre espacios vacíos, incluso si el camino entre dos espacios vacíos está en diagonal.
- Cada espacio estará separado de otro por una coma.
- Cada casilla en el mapa tendrá una coordenada. Esta estará dada por un par letra-número, en donde la primera casilla A0 corresponde a la esquina superior izquierda del mapa.

Por otro lado, el archivo se encuentra corrupto y presenta los siguientes problemas:

1. Hay símbolos aleatorios repartidos por todo el mapa. Estos deberían ser obstáculos. Cada símbolo puede ser cualquiera de estos: !"#\$%&()=?

2. Hay letras aleatorias repartidas por todo el mapa. Estas letras son información extra sin ningún significado.
3. Cada número que indica la dificultad del espacio estará expresado como una serie de operaciones aritméticas simples (sumas y restas). Por ejemplo, 2 casillas seguidas válidas pueden ser:  
 $2 + 3 - 4, 2 + 1$

Deberá ARREGLAR el archivo (guardar el mapa arreglado) antes de comenzar a trabajar con él.

## Construcción del mapa

Una vez que logre reconstruir los datos reales del mapa, deberá implementar una estructura de datos propia, que sea adecuada para representar dicho mapa, y así poder utilizar los algoritmos de navegación correctamente.

Para esto, se le pide que construya un grafo usando programación orientada a objetos, en donde cada vértice sea una celda del mapa, y además cada vértice esté conectado solamente con sus vecinos. Como restricción, para construir su grafo no podrá usar explícitamente listas ni matrices de adyacencia o incidencia.

## Consultas

Su programa debe permitir hacer consultas al mapa. Existen 3 tipos:

- **Ruta óptima.** Esta consulta debe entregar el camino óptimo, considerando los costos, desde un punto a otro del mapa presentado en una lista de las aristas. Se debe entregar el resultado usando 3 tipos distintos de heurísticas: Manhattan, distancia Euclidiana y alguna inventada por ustedes (queda completamente a su criterio, no se compliquen en ser creativos) y se debe indicar la mejor (en costo total). Un ejemplo de la consulta podría ser:

Ingrese coordenada origen: A1

Ingrese coordenada destino: A3

Camino Manhattan: (A1,A2), (A2, A3); costo 2

Camino Euclides: (A1,A2), (A2, A3); costo 2

Camino BASTIAN: (A1,B1), (B1, B2), (B2, A3); costo 10

Mejor ruta: Camino Manhattan # Hay empate, pero no importa cual de los dos entregar.

- **Obstaculo más cercano.** Esta consulta debe entregar el obstáculo más cercano desde un punto dado. Se espera que el algoritmo usado para esto tenga una complejidad de  $O(V + E)$  siendo  $V$  número de vertices y  $E$  número de aristas.<sup>1</sup> Un ejemplo de la consulta podría ser:

Ingrese coordenada origen: A1

El obstáculo más cercano se encuentra en el punto: C3

- **Ruta más corta.** Esta consulta debe entregar el camino por el cual se llega con la menor cantidad de pasos (sin considerar costos) desde un punto a otro. Un ejemplo de la consulta podría ser:

Ingrese coordenada origen: A1

Ingrese coordenada destino: A3

Camino mas corta: (A1,A2), (A2, A3)

## Informe

Deberán escribir un informe donde se hagan comparaciones de las distintas heurísticas y se mencionen los algoritmos usados para las consultas, justificando por qué se utilizó cada uno de ellos. Finalmente, deben entregar un PDF y el .zip con todo lo necesario para que el ayudante pueda compilar nuevamente el output del archivo  $\text{\LaTeX}$ .

---

<sup>1</sup>Esto es un hint.

## Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

*“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.