Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2115 – Programación como Herramienta para la Ingeniería (II/2017)

Tarea 1

Objetivos

- Modelar correctamente utilizando Programación Orientada a Objetos (POO).
- Diseñar una red de metro con las estructuras de clases vistas en clases.
- Manejar *inputs* del usuario para realizar consulta sobre la red de metro
- Redactar un informe mediante el uso de LaTeX.

Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** GitHub
- **Fecha:** 14 de septiembre
- **Hora:** 23:59
- **Desarrollo Individual**

Recomendaciones

Lea la tarea completa antes de comenzar a trabajar y fíjese en los sustantivos y verbos relacionados con el problema.

Al terminar la tarea escriba un comentario al final de su informe dando a conocer su opinión y/o su percepción de la tarea. Estos comentarios son útiles para mejor el curso durante el mismo semestre, por lo que se sugiere realizarlos a conciencia. No habrá penalizaciones en casos donde el comentario sea negativo.

Introducción

Nuestro Decano se aburrió de construir nuevos edificios en San Joaquín y ha comenzado un nuevo proyecto: un sistema de Metro que pasa por debajo del actual, exclusivo para estudiantes y funcionarios de la Universidad Católica. Para este gran proyecto se ha comenzado a discutir cual sería una red de metro eficiente para nuestro país (parece que no le gusta la red de metro chilena) y se está debatiendo entre implementar una red como la de Moscú, la de Hong Kong o la de Boston. Por esta razón, se te ha encomendado la gran misión de implementar un programa capaz de modelar estas lineas de metro y realizar una serie de consultas que serán vitales para decidir cuál línea de Metro se construirá.

Descripción

El objetivo de esta tarea es modelar y realizar una serie de consultas a una línea de Metro. Para esto, se les entregará información sobre las estaciones de Metro existentes y 3 modelos distintos de cómo están conectadas estas estaciones.

Estaciones y modelos

Junto al enunciado, se adjunta un código con 2 funciones:

- **obtener_informacion()**: esta función retorna un diccionario cuyas *keys* son los nombres de las estaciones de metro y la *value* es otro diccionario con la información de esa estación. Esta información tiene el siguiente formato

  ```
  {
      "TPE": X,
      "NPA": X,
      "NPD": X
  }
  ```

  Donde:

  - TPE: corresponde al tiempo promedio de espera en esa estación para abordar el tren.
  - NPA: corresponde al número de personas que abordan el tren.
  - NPD: corresponde al número de personas que desalojan el tren.

    Todos estos valores son constantes.

Ejemplo:

```
{
    "Baquedano": {
                    "TPE": 32,
                    "NPA": 12,
                    "NPD": 16
                },
    "Santa Lucia": {
                    "TPE": 67,
                    "NPA": 32,
                    "NPD": 11
                },
}
```

- **obtener_modelo(numero_del_modelo)**: Dado el número del modelo de una red de metro, esta función retornará una lista de tuplas de la forma (estación_1, estación_2, x_distancia, tiempo), lo cual significa que la estación_1 está conectada directamente (sin otra estación de por medio) a la estación_2 mediante una vía de x_distancia kilómetros la cual requiere de un cierto tiempo en segundos para ser recorrida. Las conexiones son bidireccionalaes por lo que puede suponer que si "X" está conectado xon "Y", entonces "Y" está conectado con "X" a la misma distancia y require del mismo tiempo para viajar de un lado al otro y viceversa.

  Ejemplo:

```
[
    ("Baquedano", "Salvador", 50, 160),
    ("Baquedano, "Parque Bustamante", 30, 200),
    ("San Joaquín", "Pedrero", 10, 60)
]
```

  En la primera linea se indica que Baquedano está conectado con Salvados a una distancia de 50 kilometros y se requiere de 160 segundos para llegar de una estación a la otra.

Consultas

Su programa debe ser capaz de recibir el nombre de dos estaciones diferentes e indicar una ruta que conecte ambas estaciones. A partir de esa ruta, su programa debe permitir realizar 5 diferentes consultas. Las primeras 2 consultas serán definidas a continuación, mientras que las otras 3 serán *consultas libres*, es decir, queda a su criterio qué tipo de consulta hacer a esa ruta.

- Transbordo: debe indicar la cantidad de transbordos existentes en esa ruta. Se define transbordo como cualquier estación que permite tomar 2 o más líneas diferentes.
- Velocidad promedio: debe mostrar la velocidad promedio a la que viaja una persona en esa ruta, para esto deberá usar la suma de los tiempos de viaje entre estaciones y la distancia total recorrida.

Interfaz (consola)

Un programa no es muy útil sin una interfaz y esta tarea no será la excepción. Para interactuar con su programa, deberá crear un menú interactivo por consola que permita realizar todas las consultas e imprima en pantalla la información obtenida de forma ordenada. Para esto, deberá solicitar *inputs* al usuario y con ello realizar las acciones necesarias para que el usuario pueda ver la información esperada. Puede asumir que el usuario no intentará destruir su programa, por lo que siempre ingresará la información de manera correcta.

Informe

Para esta tarea deberá entregar un **Manual de Usuario** escrito en **LaTeX** donde se explique cómo interactuar con la interfaz de su programa. En él, deberá mostrar cada Menú existente, cuáles son los *inputs* que pide y las acciones que desencadena cada uno. Además deberá mostrar **vía ejemplos** cómo ejecutar cada consulta implementada, junto con los resultados que ella genera. Por ejemplo, una explicacion de un Menú existente es: (Esto no es necesariamente un menú que tengan en su tarea)

```
Menú Inicial:
¿Qué desea hacer, entrar o registrarse?
```

El menú inicial pide un *string* (palabra) como *input*, que puede ser "login" o "registrarse". Las acciones a realizar son:

- login: ir al menú de ingreso.
- registrarse: ir al menú de registro.

Bonus

Se otorgará un puntaje adicional si entre las *consultas libres* escoge algunas de las siguientes (puede escoger máximo 3 consultas):

- Indicar otro par de estaciones de metro que posean la misma distancia que la ruta encontrada por su programa.
- Encontrar la vía más rápida para llegar desde una estación a otra.
- Encontrar la vía más larga para llegar desde una estación a otra, sin repetir estaciones.
- Mostrar alguna de las consultas anteriores en un ranking con tiempo, es decir, las 5 más rápidas, o 5 más largas.

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.