



1. Descripción de la Tarea

En esta tarea deberás graficar un conjunto de rutas de camiones de transporte de valores en Chile, dadas las coordenadas georeferenciales de las ciudades, y una matriz de adyacencia entre las ciudades.

2. Datos de Entrada

Tus insumos para la tarea serán dos archivos de tipo **Comma-Separated Values (CSV)**, y un último archivo de texto (TXT)¹. El primer archivo es “**ciudades.csv**” y contiene los nombres y coordenadas GPS de las ciudades con que se debe operar. Ejemplo:

```
Santiago,"-33.4569397, -70.6482697"  
Antofagasta,"-23.6523609, -70.395401"
```

Notar que la primera componente de la coordenada GPS es la latitud (eje vertical en un plano cartesiano), y la segunda es la longitud (eje horizontal).

El segundo archivo es “**adyacencia.csv**”, y contiene una matriz de adyacencia que permite determinar si existe una ruta entre un par de ciudades. Los valores en las celdas de la matriz son 1 si existe una ruta directa entre un par de ciudades, o 0 si no existe ruta directa entre el par. Ejemplo:

```
CIUDADES,Santiago,Antofagasta,Vina del Mar  
Santiago,1,0,1  
Antofagasta,0,1,0  
Vina del Mar,1,0,1
```

En este ejemplo, Santiago y Viña del Mar son alcanzables mutuamente, pero Antofagasta está aislada.

Finalmente, el último archivo es “**rutas.txt**”, el cual en cada línea contiene la ruta que debe seguir un camión. Cada ruta está compuesta por un conjunto de nombres de ciudades, y el recorrido del camión debe seguir el mismo orden con que aparecen las ciudades en la ruta. Ejemplo:

```
Santiago,Rancagua,Talca,Chillan,Concepcion
```

3. Salida del Programa

La salida del programa será mixta, pues contendrá texto en la consola y gráficos. **Para cada una de las rutas del archivo “rutas.txt”**, el programa debe hacer lo siguiente:

1. [2 puntos] Desplegar un gráfico que muestre el recorrido del camión. En este caso, el gráfico utilizará flechas (se debe usar función **arrow** de **matplotlib**²) para unir cada par de ciudades. Si una ciudad del recorrido no fuera alcanzable (matriz de adyacencia tiene un cero), se debe desplegar la ciudad no alcanzada como un punto triangular de color rojo. Además, se deben desplegar los nombres de las ciudades en el gráfico, en las coordenadas donde se encuentran las ciudades. Para eso, se debe usar la función **text** de **matplotlib**³. Se deben ajustar los ejes del gráfico para que aparezcan todas las ciudades alcanzadas, junto con la primera no alcanzable.

¹Es con formato similar a CSV, pero líneas no tienen igual número de columnas, por lo cual no es estrictamente CSV

²Ver documentación en https://matplotlib.org/api/_as_gen/matplotlib.pyplot.arrow.html

³https://matplotlib.org/api/_as_gen/matplotlib.pyplot.text.html

2. [1 punto] Imprimir la distancia total recorrida por el camión hasta la última ciudad alcanzada en la ruta.
3. [1 punto] Imprimir el consumo de combustible del camión. Para efectos del cálculo, considerar consumo de 26,5 litros por 100 kilómetros.

Finalmente, después de generar la salida anterior, se deben generar dos histogramas en un mismo plot (es decir, un plot de una fila y dos columnas de gráfico) que ilustren lo siguiente:

1. [1 punto] Un histograma para el consumo de combustible (usar `pyplot.hist`). Esto permitirá visualizar la distribución del consumo de combustible en los viajes realizados por la flota de camiones. El título del histograma debe ser “Consumo de Combustible”. El eje horizontal debe estar rotulado con “Combustible [Lt]”, y el eje vertical con “Viajes”.
2. [1 punto] Un histograma para las distancias totales de viaje. Esto permitirá visualizar la distribución de distancias de viaje para la flota de camiones. El título del histograma debe ser “Distancias de viaje”. El eje horizontal debe estar rotulado con “Distancias [Km]”, y el eje vertical con “Viajes”.

Los histogramas deben construirse considerando todas las rutas en el archivo “`rutas.txt`”.

3.1. Ejemplos

A continuación se muestra un ejemplo de la ruta Santiago ->Rancagua ->Talca ->Chillan ->Concepcion ->Temuco ->Valdivia ->Osorno ->Puerto Montt:

Ruta: Santiago, Rancagua, Talca, Chillan, Concepcion, Temuco, Valdivia, Osorno, Puerto Montt

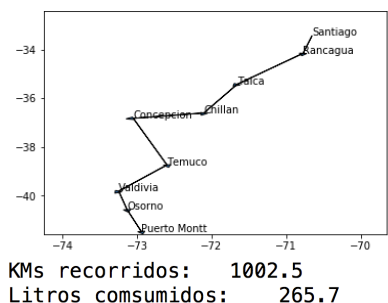


Figura 1: Ruta completamente realizable

El siguiente ejemplo muestra una ruta que no puede completarse:

Ruta: Temuco, Valdivia, Osorno, Puerto Montt, Osorno, Valdivia, Temuco, Concepcion, Talca, Rancagua, Santiago
Talca no es alcanzable desde Concepcion:

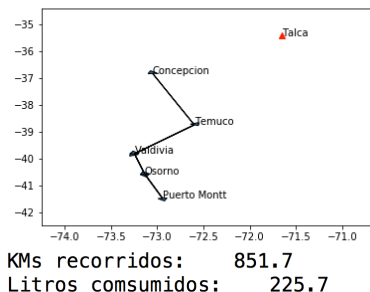


Figura 2: Ruta parcialmente realizable

Los histogramas finales debieran desplegarse de la siguiente manera:

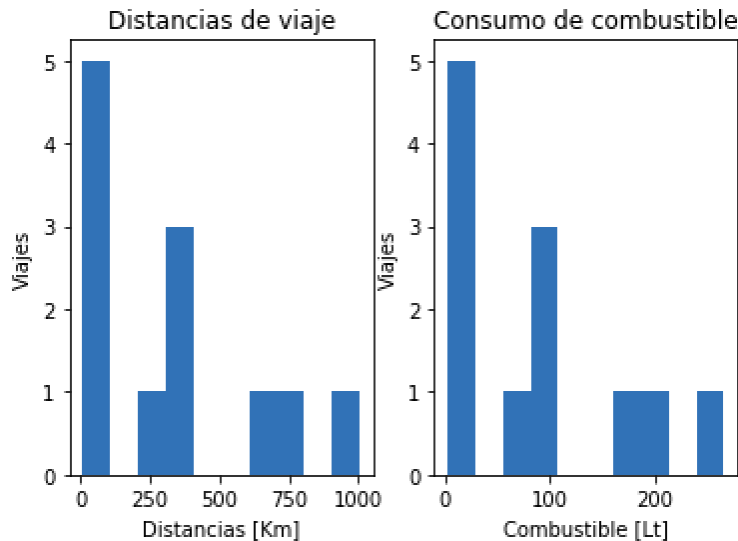


Figura 3: Despliegue de histogramas finales

4. Código Base

Se te proporcionará un archivo de código base que contendrá la función `distancia(c1,c2)`. Esta función recibe dos tuplas, cada una con coordenadas GPS. La función retorna una aproximación de la distancia en kilómetros entre las dos coordenadas GPS.

Además, el código base mostrará cómo usar `arrow` y `text` de `matplotlib`.

5. Evaluación y Entrega

Para cada ítem en este enunciado, se evaluará considerando logro parcial con ponderadores $\times 0,25$, $\times 0,5$ y $\times 0,75$.

¡Organice su código adecuadamente en funciones!

Fecha de entrega: 21 de junio de 2018 hasta las 23:59, según la hora del sistema canvasUANDES, oficial en toda la universidad. Por ningún motivo se aceptarán tareas atrasadas.

Formato de entrega: Subir **un solo archivo comprimido en formato zip** que contenga el programa Python del problema evaluado (archivo `t4.py`), al buzón en la página de la tarea en el sitio web del curso en canvasUANDES <https://uandes.instructures.com>. El nombre de archivo debe ser “TareaM4-RUT.zip”, reemplazando “RUT” por su número de RUT sin puntos ni guión. Por ejemplo, si su RUT es 18.123.122-K, el nombre de archivo debe ser “Tarea4-18123122k.zip”.

No es necesario que incluya los archivos de entrada CSV y TXT con su tarea.

Aquellas tareas que no funcionen correctamente a tal punto que no sea posible ejecutarlas, o produzcan errores que impidan la ejecución, serán calificadas con nota máxima 3,9.

Debe asegurarse de que su tarea sea descargable de forma íntegra desde canvasUANDES (el archivo `.zip` con el `t4.py` en su interior).

6. Consideraciones Éticas

El trabajo en las tareas es estrictamente individual, por lo que cuide su tarea para que no sea copiada **parcial o íntegramente** por otros. Cualquier copia será penalizada, recibiendo el mismo castigo tanto

quien copia como quien permite que le copien. También es considerada copia cualquier ayuda externa recibida directamente en la tarea, sin importar si proviene de un alumno del curso, de la universidad, o de otro lugar. El castigo será establecido por el Consejo de la Facultad, siendo como mínimo un 1,0 de promedio en el curso.