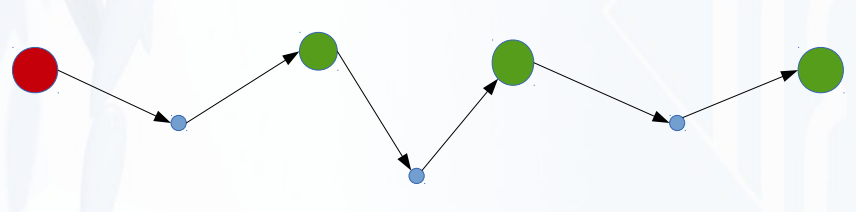
1. ***Introducción al problema y dominio: Openstreetmap y  problema de búsqueda de estados.***

En esta práctica vamos a implementar un programa capaz de establecer una ruta para salir de un punto de una calle y visitar un conjunto de lugares.



Para ello seguiremos varias estrategias de búsqueda sobre un espacio de estados para conseguir un recorrido más corto u óptimo. Las estrategias que usaremos son:

* Profundidad
* Coste Uniforme
* Anchura
* Voraz
* A\*

Posteriormente compararemos los resultados de cada estrategia para ver la complejidad de cada una y saber cuáles son las más óptimas

El espacio de estados sobre el que se delimitará la búsqueda es un trozo de mapa. Como por ejemplo un trozo de Ciudad real:



Para poder trabajar sobre este trozo de mapa utilizaremos la herramienta de la página OpenStreetMap OSM. Con esta herramienta conseguiremos un archivo osm, que contiene todas las vías y nodos de ese mapa.

Para manejar este archivo osm utilizaremos la librería osm que nos permite seleccionar los nodos y vías que queramos.

Una vez hecha la selección crearemos un grafo, que es el espacio de estados, a partir del cual haremos las correspondientes búsquedas.

***3.Estructuras y clases definidas y su justificación.***

Clase principal

En la clase principal(main) realizaremos la lectura del archivo osm. A partir de la lectura crearemos una tabla-hash con los nodos seleccionados con su respectiva longitud y latitud.

Una vez hecho esto, crearemos el grafo a partir de la tabla-hash, que nos servirá también para ponerle peso a las aristas.

En la clase principal es donde haremos el algoritmo de búsqueda para obtener la solución. Una vez obtenida la solución, creamos una función para obtener la función en un archivo gpx.

Clase EspacioEstados

En esta clase estableceremos como atributos la longitudes y latitudes máximas y mínimas del espacio de estados.

En esta clase podremos saber si un estado es válido , si se ha llegado al objetivo del problema y también podemos obtener los sucesores de un nodo. Para saber los sucesores utilizaremos el grafo utilizado en la clase principal.

Clase Estado

En esta clase únicamente crearemos los atributos del objeto estado: localización, objetivos, latitud y longitud.

Clase nodoBusqueda

En esta clase definimos los atributos del objeto nodo: id, padre, estado, costo, acción, profundidad, valor.

Clase Frontera

En esta clase creamos la lista Frontera e implementamos las funciones para crearla y modificarla.

Clase Problema

En esta clase definimos los atributos del problema: Espacio de estados y estado inicial.

Como funciones, podemos saber si un estado es el estado objetivo del problema, podemos crear la lista de nodos que añadiremos a la frontera, creamos la función para crear la solución, implementamos la función poda y la función para calcular la heurística.

DIAGRAMA UML:

