

1 Introducció

La finalitat d'aquest programa és calcular el camí més curt d'un punt d'un mapa entre dos punts. Aquest mapa es guarda en dos fitxers: `Nodes.csv` que té la id del node i la seva geolocalització (latitud, longitud), i `Carrers.csv` que té la id del carrer i les ides dels nodes que el componen.

Aquest problema hauria de poder trobar ràpidament la solució amb un mapa gran, i per això s'han fet diverses optimitzacions, com A*.

2 Sintaxi d'execució

Abans d'executar, s'ha de compilar mitjançant la següent comanda:

```
cc AEstrella.c -o AEstrella -Wall -lm
```

i ara que està compilat, es pot executar. Per executar aquest programa s'ha de posar la següent comanda a la consola estant a la carpeta on es té `Carrers.csv` i `Nodes.csv`:

```
./AEstrella id_node_inici id_node_final
```

3 Explicació de l'estructura del codi

Primer s'importen les llibreries i es defineixen constants (linies 3-9)

Després es defineixen les estructures adjacents (linies 11-33), podem veure com es busca que estructures com `INFO_ESTRELLA` tinguin un apuntador a l'estructura `NODE` i que també `NODE` pugui apuntar a `INFO_ESTRELLA`. Per això està el typedef de la línia 11.

Seguidament es defineixen les funcions necessàries (línia 35-114):

1. `esferiquesACartesians` (línia 35-39): Pasa la latitud i longitud d'un node a coordenades cartesianes.
2. `encuaAmbPrioritat` (línia 41-64): Mitjançant un node i l'inici de la cua, la funció encua amb prioritat tenint en compte la distància i la funció heurística.
3. `extreureDeLaCua` (línia 66-81): També fent servir un node i l'inici de la cua treu aquest node de la cua.
4. `distancia` (línia 83-95): Calcula la distància entre dos nodes, retorna aquesta. Fa servir la funció `esferiquesACartesians` i `sqrt` (aquesta última definida a `<math.h>`)
5. `buscaPosicioNode` (línia 97-114): Fa servir cerca binària per trobar la posició d'un node mitjançant la seva id.

Finalment, tenim el main (linies 116-270):

1. Declaració de totes les variables necessaries (linies 117-131).
2. Lectura de `Nodes.csv` (linies 134-155).
3. Lectura de `Carrers.csv` (linies 158-184).
4. L'algorisme A* (linies 187-235).
5. Mostrar el camí (linies 238-269).

4 Exemple de funcionament

Per exemple

```
./ AEstrella 0259184345 1793441250
```

en el meu cas, retorna:

```
# La distancia de 259184345 a 1793441250 es de 507.88680 metres.
```

```
# Cami optim:
```

Id=0259184345		41.545380		2.106830		Dist=0.00000
Id=0259437888		41.545752		2.106744		Dist=42.04203
Id=0259437905		41.546388		2.106495		Dist=115.72240
Id=0259438253		41.546734		2.107858		Dist=235.47656
Id=0965459173		41.546963		2.107746		Dist=262.61711
Id=0960085142		41.547169		2.107648		Dist=286.90038
Id=1944921315		41.547281		2.107553		Dist=301.62308
Id=2412854895		41.547777		2.107092		Dist=368.87952
Id=1944921533		41.548161		2.107668		Dist=433.05849
Id=1944921536		41.548240		2.107798		Dist=446.89987
Id=1944921547		41.548280		2.107864		Dist=454.04708
Id=1793441253		41.548396		2.108055		Dist=474.49589
Id=1944921549		41.548399		2.108073		Dist=476.04126
Id=1955175329		41.548407		2.108110		Dist=479.20011
Id=1955175330		41.548452		2.108329		Dist=498.15467
Id=1793441250		41.548481		2.108440		Dist=507.88680

```
# -----
```