

Matemàtica computacional i analítica de dades

ALGORISMIA I COMBINATORIA EN GRAFS

PRACTICA INDIVIDUAL

Biel González Garriga
NIU: 1551813

Curs 2020-21

1 Introducció al problema

Tenim un graf format per nodes i arestes que representa el mapa real d'una zona. A partir de les dades d'aquest graf volem implementar l'algorisme A* per trobar el camí més curt entre dos nodes donats.

2 Com executar el codi

Primer de tot has de situar-te al directori on tens el arxiu "AEstrella.c", després s'ha de compilar el codi des de terminal amb l'ordre:

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

Un cop s'hagi generat l'executable amb el nom "AEstrella" ja pots executar el programa. Per executar correctament el codi s'han d'introduir dos arguments, el node d'inici i el node final, tal que així:

```
./AEstrella nodeInici nodeFinal
```

O en el cas de que s'executés a Windows:

```
AEstrella nodeInici nodeFinal
```

3 Main i funcions

3.1 Funcions

Una part de les funcions s'han extret d'[aquí](#) i s'han adaptat per el seu correcte funcionament amb les variables del programa.

- `double distancia(node, node)`
És la funció heurística del A*, calcula la distancia entre dos nodes en línia recta. Retorna la distancia entre els dos nodes.
- `unsigned BuscaPunt(long int, node*, int)`
Busca un node al vector de nodes. Retorna el index del node al vector.
- `unsigned ExtractMin(PriorityQueue *)`
Extreu el primer element de la cua de prioritat, que será el de f mínima. Retorna el index del node al vector de nodes.

- `void RequeueWithPriority(unsigned, PriorityQueue *, AStarControlData *)`
Torna a encuar un element que ja estava amb anterioritat a la cua.
- `unsigned AddWithPriority(unsigned, PriorityQueue *, AStarControlData *)`
Encua amb prioritats a la cua un node. Retorna 1 si s'ha fet correctament i 0 si no.
- `int IsEmpty(PriorityQueue)`
Mira si la cua es buida. Retorna 0 si no ho és, i qualsevol altre valor int si ho és.
- `void ExitError(const char *miss, int errcode)` S'encarrega forçar la sortida del programa amb un determinat codi d'error que se li passa com `errcode`.
- `unsigned AStar(node *, AStarPath *, unsigned, unsigned, unsigned)`
Implementa el algorisme A*. Rep un apuntador al vector de nodes, un apuntador al vector que contindrà el camí, el nombre de nodes del graf, el index del node d'inici i el index del node final. Al finalitzar l'execució de la funció, si tot ha anat amb normalitat, tindrem el camí fet mitjançant A* omplert amb el camí que buscàvem.
La funció reserva l'espai necessari per el vector que contindrà la informació de si un node està obert o tancat, i posa els valor corresponents al node inicial, g, qui es el pare i f, després encua amb prioritats. Llavors comença el bucle principal de la funció, on es desencua el primer node de la llista de prioritats i es miren els nodes adjacents a aquest. Es calculen la g i la f del node i després si està tancat se l'encua amb prioritats i si està obert se'l reencua amb prioritats. Aquest bucle es durà a terme fins que, o la cua estigui buida o s'hagi arribat al node objectiu.
Retorna un -1 si no s'ha pogut reservar memòria per la llista de nodes oberts, 0 si no s'ha trobat cap camí i 1 si existeix un camí.

3.2 Main

Al main es duen a terme quatre tasques, la lectura d'arguments que se li passen al programa, la lectura d'arxius, l'execució de la funció AStar i l'impresió del camí, tot en aquest mateix ordre. Es controlen possibles problemes de assignació de memòria als vectors de nodes i carrers, que no es pugui obrir correctament els arxius, que els nodes introduïts no existeixin i els valors d'error que podria retornar la funció AStar.

4 Exemple de fonctionnement

Executem el programa tal que així:

```
./AEstrella 259184345 1793441250
```

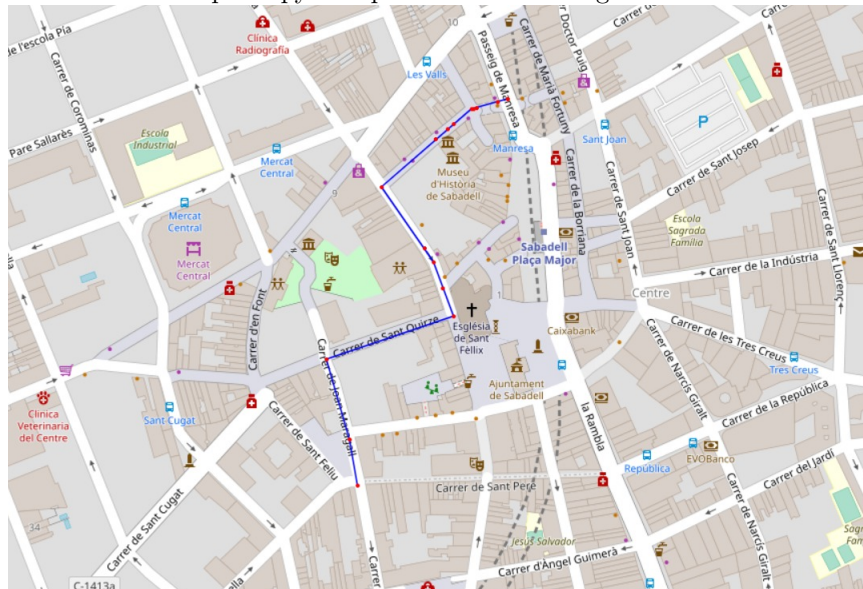
El programa farà el print per pantalla següent en linux:

```

linuxuser@linuxuser-VirtualBox:~/Baixades$ gcc AEstrella.c -o AEstrella -Wall -lm
linuxuser@linuxuser-VirtualBox:~/Baixades$ ./AEstrella 0259184345 1793441250
# La distancia de 259184345 a 1793441250 es de 507.886803 metres.
# Cami optim:
Id=0259184345 | 41.545380 | 2.106830 | Dist = 0.000000
Id=0259437888 | 41.545752 | 2.106744 | Dist = 42.042028
Id=0259437905 | 41.546388 | 2.106495 | Dist = 115.722403
Id=0259438253 | 41.546734 | 2.107858 | Dist = 235.476557
Id=0965459173 | 41.546963 | 2.107746 | Dist = 262.617110
Id=0960085142 | 41.547169 | 2.107648 | Dist = 286.900380
Id=1944921315 | 41.547281 | 2.107553 | Dist = 301.623075
Id=2412854895 | 41.547777 | 2.107092 | Dist = 368.879521
Id=1944921533 | 41.548161 | 2.107668 | Dist = 433.058491
Id=1944921536 | 41.548240 | 2.107798 | Dist = 446.899874
Id=1944921547 | 41.548280 | 2.107864 | Dist = 454.047082
Id=1793441253 | 41.548396 | 2.108055 | Dist = 474.495886
Id=1944921549 | 41.548399 | 2.108073 | Dist = 476.041257
Id=1955175329 | 41.548407 | 2.108110 | Dist = 479.200109
Id=1955175330 | 41.548452 | 2.108329 | Dist = 498.154671
Id=1793441250 | 41.548481 | 2.108440 | Dist = 507.886803
#
-----
linuxuser@linuxuser-VirtualBox:~/Baixades$

```

Si executem l'script de python per veure el recorregut ens dona:



5 Extra

Aquest codi també es pot executar a Windows, però no treu sempre el camí més curt entre dos nodes, fent servir el mateix exemple que a la secció anterior ens dona que:

```
C:\Users\Biel GG\CLionProjects\untitled7>gcc -o AEstrella AEstrella.c -Wall -lm

C:\Users\Biel GG\CLionProjects\untitled7>AEstrella 0259184345 1793441250
# La distancia de 259184345 a 1793441250 es de 523.487238 metres.
# Cami optim:
Id=0259184345 | 41.545380 | 2.106830 | Dist = 0.000000
Id=0259437888 | 41.545752 | 2.106744 | Dist = 42.042028
Id=0259437889 | 41.545797 | 2.107053 | Dist = 68.237554
Id=0259437890 | 41.545903 | 2.107729 | Dist = 125.763074
Id=0259437805 | 41.545943 | 2.108129 | Dist = 159.279838
Id=1945672908 | 41.545959 | 2.108187 | Dist = 164.467884
Id=1945672910 | 41.545975 | 2.108265 | Dist = 171.185696
Id=1945672929 | 41.546072 | 2.108709 | Dist = 209.702232
Id=0259437827 | 41.546080 | 2.108750 | Dist = 213.234013
Id=1946430351 | 41.546096 | 2.108798 | Dist = 217.526695
Id=1946430357 | 41.546184 | 2.109108 | Dist = 245.176915
Id=0259437891 | 41.546222 | 2.109170 | Dist = 251.882049
Id=0965631475 | 41.546576 | 2.108988 | Dist = 293.996213
Id=0965631476 | 41.546758 | 2.108946 | Dist = 314.619968
Id=0965631457 | 41.546853 | 2.108958 | Dist = 325.187095
Id=1389700939 | 41.546917 | 2.108953 | Dist = 332.326345
Id=1389701038 | 41.547476 | 2.108898 | Dist = 394.628591
Id=1449178113 | 41.547900 | 2.108861 | Dist = 441.851866
Id=0255402708 | 41.548108 | 2.108818 | Dist = 465.243897
Id=0965193247 | 41.548096 | 2.108781 | Dist = 468.536228
Id=0255402709 | 41.548158 | 2.108599 | Dist = 485.245670
Id=1793441250 | 41.548481 | 2.108440 | Dist = 523.487238
# -----
```

