PRÁCTICA 1 BÚSQUEDAS

PARTE 1: BÚSQUEDA NO INFORMADA. RAMIFICACIÓN Y ACOTACIÓN

Para la elaboración de esta tarea, se ha utilizado el código base de la práctica, en la cual se ha añadido un gestor de colas, denominado **"gestorBab"** en el archivo ***utils.py***, reutilizando el gestor del cual ya disponíamos en el código. Este nuevo **"gestorBab"** tiene una pequeña diferencia con respecto al anterior, en la función **extend()** (función que añade los elementos), se ha introducido la función **.sort()** la cual ordena dichos elementos en función del **path_cost**, lo cual hace posible que se acumule el coste de cada ruta para encontrar la solución

PARTE 2: BÚSQUEDA INFORMADA. RAMIFICACIÓN Y ACOTACIÓN CON SUBESTIMACIÓN

En esta parte, estamos ante un método de búsqueda informada, denominado ramificación y acotación con subestimación, y pertenece a una búsqueda informada, debido a que se introduce una subestimación lo cual se considera una heurística (Una línea recta que une el nodo inicial con el final).
Para esta segunda parte, se ha añadido al código de la parte anterior otro gestor de colas, denominado **"gestorBab2"** en el archivo ***utils.py***, el cual realiza la misma función pero con una pequeña diferencia y es que en el método que añade elementos (**extend()**) se ordenan los elementos con la función **.sort()** teniendo en cuenta el **path_cost** y **la heurística** (se indica como **self.problem.h(x)**)

Ambos deben dar el mismo resultado, pero el método informado efectuará menos saltos que el no informado. En ambos se ha establecido una variable "count" en **graph_search que almacena el número de saltos de los métodos**

RESULTADOS

Arad-Bucharest

```
ab = search.GPSProblem('A', 'B'
                       , search.romania)

print('Búsqueda anchura')
print(search.breadth_first_graph_search(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Búsqueda profundidad')
print(search.depth_first_graph_search(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Rafimifación y salto')
print(search.bab(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Rafimifación y salto con Subestimacion')
print(search.babSub(ab).path())
```

```
C:\Users\Belen\PycharmProjects\prac1fsi\venv\Scripts\python.exe "C:/Users/Belen/Desktop/universidad/FSI/practicas terminadas/prac1FSI/run.py"
Búsqueda anchura
N° de saltos = 16
[<Node B>, <Node F>, <Node S>, <Node A>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Búsqueda profundidad
N° de saltos = 10
[<Node B>, <Node P>, <Node C>, <Node D>, <Node M>, <Node L>, <Node T>, <Node A>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Rafimifación y salto
N° de saltos = 24
[<Node B>, <Node P>, <Node R>, <Node S>, <Node A>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Rafimifación y salto con Subestimacion
N° de saltos = 6
[<Node B>, <Node P>, <Node R>, <Node S>, <Node A>]

Process finished with exit code 0
```

Arad-Craiova

```
ab = search.GPSProblem('A', 'C'
                       , search.romania)

print('Búsqueda anchura')
print(search.breadth_first_graph_search(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Búsqueda profundidad')
print(search.depth_first_graph_search(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Rafimifación y salto')
print(search.bab(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Rafimifación y salto con Subestimacion')
print(search.babSub(ab).path())
```

```
C:\Users\Belen\PycharmProjects\prac1fsi\venv\Scripts\python.exe "C:/Users/Belen/Desktop/universidad/FSI/practicas terminadas/prac1FSI/run.py"
Búsqueda anchura
N° de saltos = 18
[<Node C>, <Node R>, <Node S>, <Node A>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Búsqueda profundidad
N° de saltos = 7
[<Node C>, <Node D>, <Node M>, <Node L>, <Node T>, <Node A>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Rafimifación y salto
N° de saltos = 20
[<Node C>, <Node R>, <Node S>, <Node A>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Rafimifación y salto con Subestimacion
N° de saltos = 7
[<Node C>, <Node R>, <Node S>, <Node A>]

Process finished with exit code 0
```

Craiova-Mehadia

```
ab = search.GPSProblem('C', 'M'
                       , search.romania)

print('Búsqueda anchura')
print(search.breadth_first_graph_search(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Búsqueda profundidad')
print(search.depth_first_graph_search(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Rafimifación y salto')
print(search.bab(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Rafimifación y salto con Subestimacion')
print(search.babSub(ab).path())
```

```
C:\Users\Belen\PycharmProjects\prac1fsi\venv\Scripts\python.exe "C:/Users/Belen/Desktop/universidad/FSI/practicas terminadas/prac1FSI/run.py"
Búsqueda anchura
N° de saltos = 5
[<Node M>, <Node D>, <Node C>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Búsqueda profundidad
N° de saltos = 19
[<Node M>, <Node L>, <Node T>, <Node A>, <Node Z>, <Node O>, <Node S>, <Node F>, <Node B>, <Node P>, <Node C>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Rafimifación y salto
N° de saltos = 5
[<Node M>, <Node D>, <Node C>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Rafimifación y salto con Subestimacion
N° de saltos = 3
[<Node M>, <Node D>, <Node C>]

Process finished with exit code 0
```

Oradea-Bucharest

```
ab = search.GPSProblem('O', 'B'
                       , search.romania)

print('Búsqueda anchura')
print(search.breadth_first_graph_search(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Búsqueda profundidad')
print(search.depth_first_graph_search(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Rafimifación y salto')
print(search.bab(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Rafimifación y salto con Subestimacion')
print(search.babSub(ab).path())
```

```
C:\Users\Belen\PycharmProjects\prac1fsi\venv\Scripts\python.exe "C:/Users/Belen/Desktop/universidad/FSI/practicas terminadas/prac1FSI/run.py"
Búsqueda anchura
N° de saltos = 14
[<Node B>, <Node F>, <Node S>, <Node O>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Búsqueda profundidad
N° de saltos = 22
[<Node B>, <Node P>, <Node R>, <Node S>, <Node O>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Rafimifación y salto
N° de saltos = 20
[<Node B>, <Node P>, <Node R>, <Node S>, <Node O>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Rafimifación y salto con Subestimacion
N° de saltos = 7
[<Node B>, <Node P>, <Node R>, <Node S>, <Node O>]

Process finished with exit code 0
```

Sibiu-Pitesti

```
ab = search.GPSProblem('S', 'P'
                       , search.romania)

print('Búsqueda anchura')
print(search.breadth_first_graph_search(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Búsqueda profundidad')
print(search.depth_first_graph_search(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Rafimifación y salto')
print(search.bab(ab).path())
print('>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>')
print('Rafimifación y salto con Subestimacion')
print(search.babSub(ab).path())
```

```
C:\Users\Belen\PycharmProjects\prac1fsi\venv\Scripts\python.exe "C:/Users/Belen/Desktop/universidad/FSI/practicas terminadas/prac1FSI/run.py"
Búsqueda anchura
N° de saltos = 15
[<Node P>, <Node R>, <Node S>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Búsqueda profundidad
N° de saltos = 3
[<Node P>, <Node R>, <Node S>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Rafimifación y salto
N° de saltos = 7
[<Node P>, <Node R>, <Node S>]
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Rafimifación y salto con Subestimacion
N° de saltos = 3
[<Node P>, <Node R>, <Node S>]

Process finished with exit code 0
```

Como podemos observar, ambos procedimientos ofrecen el mismo resultado, pero el método de ramificación y acotación con subestimación realiza menos saltos gracias a la heurística introducida.