PROGRAMMING PROJECT II

LUIZ HENRIQUE MAMEDE QUEIROZ

PEDRO HENRIQUE RIBEIRO DE OLIVEIRA

RAPHAEL MORAGAS DIAS TAVEIRA GONÇALVES

Delivery Routing Algorithms

Choose dataset

- Choose between the toy graphs, the medium graphs, the real-world graphs, and a graph provided by yourself

- Each of the classes above (except the custom) have their own submenu

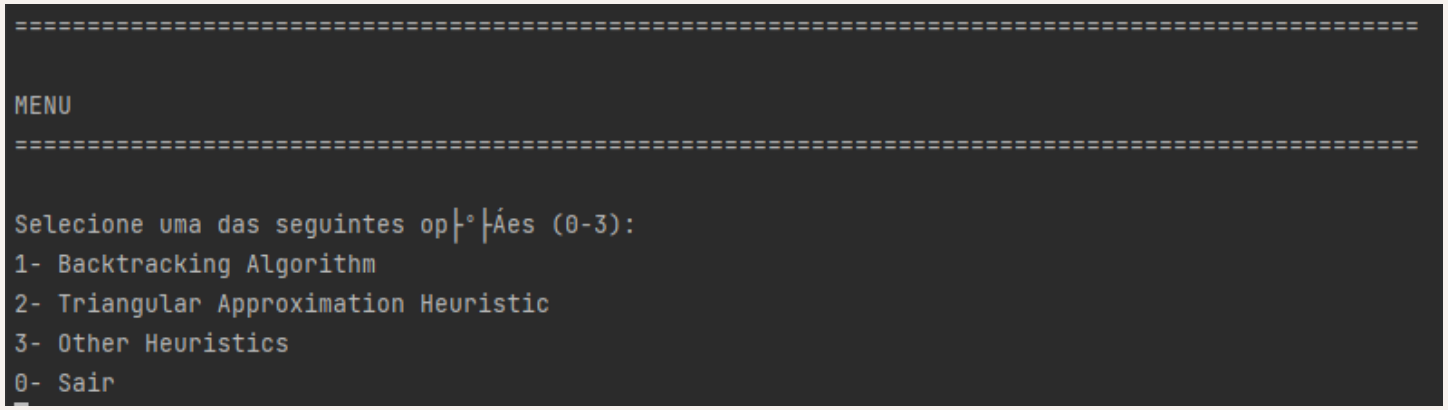
```
================================================================================================
                                                                                SETTING GRAPHS
================================================================================================

Digite uma das seguintes op├°├Áes (0-4) de acordo com o tipo de grafo deseja utilizar:
1- Toy Graphs
2- Extra Medium-Size Graphs
3- Real-World Graphs
4- Fornecer um grafo diferente (com nomes edges.csv e nodes.csv)
0- Sair
```

```
Digite uma das seguintes op├°├Áes (0-3) para escolher o Toy Graph:
1- Shipping
2- Stadiums
3- Tourism
0- Sair
```

User interface

- Choose between the 4 functions implemented
- Other Heuristics opens a menu with 2 options
- Input 0 allows you to choose a different set to use a different dataset



1-Backtracking Algorithm

Backtracking Algorithm

- Too computationally heavy to be viable in larger graphs
- Guarantees most optimal path

- A "smart bruteforce"
- Recursive
- Tests every possible path
- On the first run saves the path and, length of tour
- If a route is shorter than the saved, update it
- Checks at each node if it has passed the saved value, if it does, returns to previous and keeps going
- At the end returns the value,

2-Triangular Approximation Heuristic

Triangular Approximation Heuristic

- Worst case scenario will be 2x the size the optimal path
- A lot faster than the backtracking algorithm
- Useful for small, medium and real-world graphs

- Computes MST from selected root using Prim's algorithm
- Do a preorder walk tour along the MST
- Return to the root completing the tour

3-Other Heuristics

Heuristic Approximation

- The fewer cycle are formed, the closer the approximation will be
- A lot faster than the backtracking approach

- Finds MST with Prim's Algorithm
- Traverse graph nodes by order of creation in MST and attempts to connect through the MST path
- If that path has already been used, looks for next shortest edge available
- A tour is made to check for possible cycle that may have been created
- If one is found, minimum cost connections are made to exit them

Christofides Algorithm

- Worst case scenario will be 1.5x the size the optimal path
- A lot faster than the backtracking approach

- Finds MST with Prim's Algorithm
- Finds all odd-degree vertices in the tree
- Finds their minimum-weight perfect match for those vertices
- Combine the edges of the tree and the matches
- Form a Eulerian circuit in the resulting graph
- Make the previous circuit into a Hamiltonian circuit by skipping repeated vertices