Laboratory practice No. 4: Greedy Algorithms

Sarah Henao Gallego
Universidad EAFIT
Medellín, Colombia
shenaog@eafit.edu.co

October 1, 2018

1) Code uploaded to GitHub

1.a. Implementation of Greedy Algorithm in Travel Salesman Problem

– The code is uploaded in GitHub. The algorithm is the same algorithm shown in Section 4, problem 2. The algorithm is not cited because there are no authors stated in the PDF file.

2) Online Exercise

–

3) Project-type Questions

3.a. What data structure did you use to solve 1.1 and how does the algorithm work?

– The data structures used in problem 1.1 are stacks and adjacency matrices. The algorithm is the Nearest neighbor algorithm and it works as follows: The algorithm starts at the first location stated in the input adjacency matrix and repeatedly visited until all have been visited.

3.b. Do Greedy Algorithms always return the optimal solution to the Traveling Salesman Problem? How should the graph be structured in order for the algorithm to at least return feasible solution? Why?

– Greedy algorithms almost always returns a solution to the TSP, however it does not always return the optimal solution. In order for the algorithm to return a solution, the route that the salesman is going to take must exist at least in one way and the algorithm must be composed of criteria that will lead to a feasible solution. This solution must satisfy the algorithm's selection criteria.

Vertex	Cost	Last Visited	Route from Vertex A
A	0	-1	A
B	20	A	A - B
C	40	F	A - B - F - C
D	50	C	A - B - F - C - D
E	Inf	-1	No route
F	30	B	A - B - F
G	**70**	**D**	**A - B - F - C - D - G**
H	60	C	A - B - F - C - H

Table 1: Shortest Routes to each vertex departing from initial node A

3.c. What data structure did you use to solve 2 and how does the algorithm work?

–

3.d. Calculate the complexity for 2.

–

3.e. What do the variables 'n' and 'm' mean in 3.4?

–

4) Practice Test Problems

4.a. SELECTING ACTIVITIES PROBLEM

4.1.1 Last Line

```
i++;
```

4.b. TRAVELING SALESMAN PROBLEM

4.2.1 Line 18

```
if ( min > adjacencyMatrix[element][i])
```

4.c. DIJKSTRA ALGORITHM

4.3.1 *Shortest Routes* departing from A to all other Vertices B, C, D, E, F, G, H

See Table 1

Profoessor Mauricio Toro Bermudez
Phone: (+57)(4)2619500 Ext. 9473. Office: 19 – 627
E-mail: mtorobe@eafit.edu.co

4.3.2 Shortest route from A to G

```
A - B - F - C - D - G  with a cost of 70.
```

4.d. VITAMINES, PROTEINS, MINERALS PROBLEM

4.4.1 Line 10

```
temp = temp/2;
```

4.4.2 Line 11

```
return temp + minimo;
```

4.4.3 Find the complexity of the algorithm

```
O(1)
```

Profoessor Mauricio Toro Bermudez
Phone: (+57)(4)2619500 Ext. 9473. Office: 19 – 627
E-mail: mtorobe@eafit.edu.co