Laboratory practice No. 3: BackTracking

Santiago Ochoa Castaño
Universidad Eafit
Medellín, Colombia
sochoac1@eafit.edu.co

Miguel Ángel Zapata Jimenez
Universidad Eafit
Medellín, Colombia
mazapataj@eafit.edu.co

3) Practice for final project defense presentation

3.1 Another way that solves this problem is by using the greedy algorithm. It consists in a method that follows the problem-solving heuristic of making the locally optimal choice at each stage. In this case, the main idea is to traverse each node by asking which successor has the lowest cost and choose that one to explore.

3.2 At the moment of listing paths, there is a kind of pattern. For example, if there is a graph with tree nodes (a,b,c) and all nodes are connected this will be the number of paths: abc, acb, bca, bac, cba, cab, ab, ba, ca, ac, bc y ac. There are 12 paths. 6 (equal to 1*2*3) where the 3 nodes are traversed without repeating and 6 where only 2 nodes are visited. Another example is with 4 nodes or vertex, in this case there is 24 (equal to 1*2*3*4) paths. Through this can be seen that there is **n!** possible paths in a graph where all nodes are connected.

3.3

N	Execution time (ms) BackTracking	Execution time(ms) Brute Force
4	2	3
5	3	5
6	1	31
7	6	84
8	5	413
9	10	10647
10	40	202280
11	98	(More than the expected time) …
12	496	
13	2492	
14	15588	
15	102934	
16	758746	

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

3.4 To traverse graphs problems is more convenient DFS when the priority is to find the most optimal path, because in that way it minimizes the time of searching. On the other hand, BFS is more functional to find all the possible routes. Additionally, BFS is more useful to find close friends in a social network because DFS would find the most distant friends at first being a big problem.

3.5 The solution to solve the problem 2.1 was possible using a data structure known as pair which allows to store a string as key that represents the route and an integer as value that determines the minimum path cost. The algorithm's functionality is based on improving the route and cost as it finds a better one. Also, it avoids going on routes that are not convenient at comparing it with the best route of each iteration.

3.6 The complexity of the algorithm 2.1 is O(V+E) since each node and edge must be traversed in the worst case.

3.7 The variables present in the complexity are V and E. V represents the number of vertex or nodes and E represents the number of edges in the graph.

3.8 The algorithm's functionality is based on improving the route and cost as it finds a better one. In first instance, it was implemented an algorithm based in DFS to determinate if there was a route between two nodes. Next, there was a conditional that compared the current cost with the best one so far. These two implementations had the objective to avoid exploring a node which didn´t led to the solution. The stop case was at exploring the arrival node. Then, it´s compared if the current cost is less than the best one so far. If it is true, both are replaced.

4) Practice for midterms

4.1

4.1.1 int res = solucionar(n-a, a, b, c);
4.1.2 res = Math.max(res, solucionar(n-b, a, b, c);
4.1.3 res = Math.max(res, solucionar(n-c, a, b, c);

4.2

4.1.1 graph.length
4.1.2 sePuede(v, graph, path, pos)
4.1.3 cicloHamiAux(graph, path, pos+1)

4.5

4.5.1 return 1 + lcs(i-1, j-1, s1, s2)
4.5.2 return Math.max(ni,nj);
4.5.3 2*T(n-1)

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

4.7

4.7.1 if(r<0)
4.7.2 a[r] = i;
4.7.3 sol(a, r-1)

References:

❖ https://en.wikipedia.org/wiki/Greedy_algorithm

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473