Laboratory practice No. 3: Backtracking

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

October 25, 2018

1) Code uploaded to GitHub

1.a. Implementation of a Backtracking Algorithm that finds a solution to the N Queen Problem

– The code is uploaded in GitHub. The file uploaded to GitHub is called **Backtracking.java** includes an algorithm to solve the N-Queens Problem using Backtracking.

1.b. Implementation of a method that determines if a given graph has cycles or not.

–

2) Online Exercise

–

3) Project-type Questions

3.a. Other than brute force and backtracking, what are other computational techniques that can be used to solve the Shortest Routing Problem?

– Another computational techniqe that can be used to solve the Shortes Routing Problem are **Greedy Algorithms**. A Greedy algorithm uses heuristic decision making to find a locally optimal choice at each stage of the algorithm with hopes of eventually reaching the global optimum. Unfortunately, these algorithms do not guarantee an optimal solution.

3.b. When is it more convenient to use DFS and BFS when traveling through graphs?

– Generally, we should use **BFS** when we want to find the shortest path from a certain node i to another node j or the minial number of steps to get from node i to node j. On the other hand, it is more convenient to use **DFS** when you want to find all the possible combinations of a graph and compare between all of them to find the best solution to your problem. Lastly, if the purpose of your algorithm is to just check if two nodes are connected or, in other words, if you can reach a node j parting from a certain node i then either DFS or BFS are useful.

3.c. Other than BFS and DFS, what are other search algorithms for graphs?

– Just like BFS and DFS, **Dijkstra's shortest path algorithm** can be used to find the shortest path between an initial node and a destination.

3.d. Explain what data structures you used to solve 2.

–

3.e. Calculate the complexity for 2.

–

3.f. What do the variables 'n' and 'm' mean in 3.5?

–

3.g. Explain the code in 1.1. How is it implemented? What data structures and algorithms were used?

– The data structures that were used are **arrays**. The algorithm was implemented **recursively**.

4) Practice Test Problems

4.a. GIVEN A SET OF A, B, C NUMBERS, FIND THE LARGEST AMOUNT OF NUMBERS TO ADD UP THAT WILL ADD UP TO N

4.1.1 Line 4

```
int res = solucionar(n-a, a, b, c) + 1;
```

4.1.2 Line 5

```
res = Math.max(res, solucionar(n-b, a, b, c));
```

4.1.3 Line 6

```
res = Math.max(res, solucionar(n-c, a, b, c));
```

4.b. HAMILTONIAN PATH

4.2.1 Line 2

```
if (pos == graph.length)
```

4.2.2 Line 9

```
if (sePuede(v, graph, path, pos))
```

4.2.3 Line 11

```
if (cicloHamilAux(graph, path, pos + 1))
```

4.c. DFS AND BFS

4.3.1 DFS

```
0, 3, 7, 4, 2, 1, 5, 6
```

4.3.2 BFS

```
0, 3, 4, 7, 2, 1, 6, 5
```

4.d. GUGOL MAS

–

4.e. LONGEST COMMON SUB-SEQUENCE

4.5.1 Line 7

```
return 1 + lcs(i-1, j-1, s1, s2);
```

4.5.2 Line 11

```
return Math.max(ni, nj);
```

4.5.3 Worst case scenario

```
T(n) = 2^n instructions
```

4.f. SEARCH ALGORITHMS DFS AND BFS FOR GRAPHS

4.6.1 A possible DFS route:

```
0, 1, 4, 3, 2
```

4.6.2 A possible BFS route:

```
0, 1, 2, 3, 4
```

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