ESTRUCTURA DE DATOS 2
Código ST0247

Laboratory practice No. 3: Backtracking

Daniel Felipe Gómez Martínez
Universidad Eafit
Medellín, Colombia
dfgomezm@eafit.edu.co

Cesar Andrés García Posada
Universidad Eafit
Medellín, Colombia
cagarciap@eafit.edu.co

Daniel García García
Universidad Eafit
Medellín, Colombia
dgarciag@eafit.edu.co

3) Practice for final project defense presentation

3.1

To find the shortest path in a graph we have methods such as Dijkstra, Greedy algorithms, Prim, Ford. These methods make it possible to find the shortest path of a node from one graph to another quickly and efficiently.

3.2

$2^{(n-1)}$ Path, where n means the number of nodes.

3.3
3.4

Examples

- **To determine which is the best possibility among all the possibilities you have in the graph, it is convenient to work with DFS**
- **To find the shortest path between nodes, or the minimum cost in a path between certain nodes, it is convenient to work with BFS.**
- **To determine if there is a path between two nodes, you can work with both methods (BFS and DFS)**
- **Finally, if the solutions are close to the root node, it is convenient to work with BFS**

3.5

The data structure implemented for the solution of Exercise 2.1 was an array where the shortest path from 1 to vertex n was saved. Regarding the operation of the code, it is based on a recursive method where a path up to the vertex is sought from the successors of the origin (1) and as we add the vertices to the matrix, the cost of the journey is also compared with in order to stop the search for the path from a vertex k which exceeds a path already found.

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

3.6

Due to the implementation of a cycle within a recursive method in the worst case the complexity of the code is O (n2).

3.7

The array 'i' always starts from one [1, 'L', 'P', ..., n].

Which 'L' is going to make a successor of 1 and P a successor of 'L' and until finding a path to n (Whenever and whenever possible).

The variable "com" will start in an Integer.MAXVALUE but it will change as a path to vertex n is found, this variable will serve to minimize the search of a path, because if a path is greater than this one, it will find new path.

So the complexity of the algorithm resembles O (n2), because if the vertices have many successors would make the process very expensive in terms of memory and efficiency.

4) Practice for midterms

4.1.1 int res = solucionar(n-1, a, b, c) + 1;
4.1.2 res = Math.max(a, b);
4.1.3 res = Math.max(res, c);
4.2.1 if (pos == 1){
4.2.2 if (sePuede(v ,graph[][] ,path[] ,pos)) {
4.2.3 if (cicloHamilAux(graph[][] ,path[] , v))
4.3.1

Iniciando en 0: 0, 3, 7, 4, 2, 1, 5, 6
Iniciando en 1: 1, 0, 3, 7, 4, 2, 6, 5
Iniciando en 2: 2, 1, 0, 3, 7, 4, 5, 6
Iniciando en 3: 3, 7
Iniciando en 4: 4, 2, 1, 0, 3, 7, 5, 6
Iniciando en 5: 5
Iniciando en 6: 6, 2, 1, 0, 3, 7, 4, 5
Iniciando en 7: 7

4.3.2

Iniciando en 0: 0, 3, 4, 2, 7, 1, 6, 5
Iniciando en 1: 1, 0, 2, 5, 3, 4, 6, 7
Iniciando en 2: 2, 1, 4, 6, 0, 5, 3, 7
Iniciando en 3: 3,7

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

Iniciando en 4: 4, 2, 1, 6, 0, 5, 3, 7

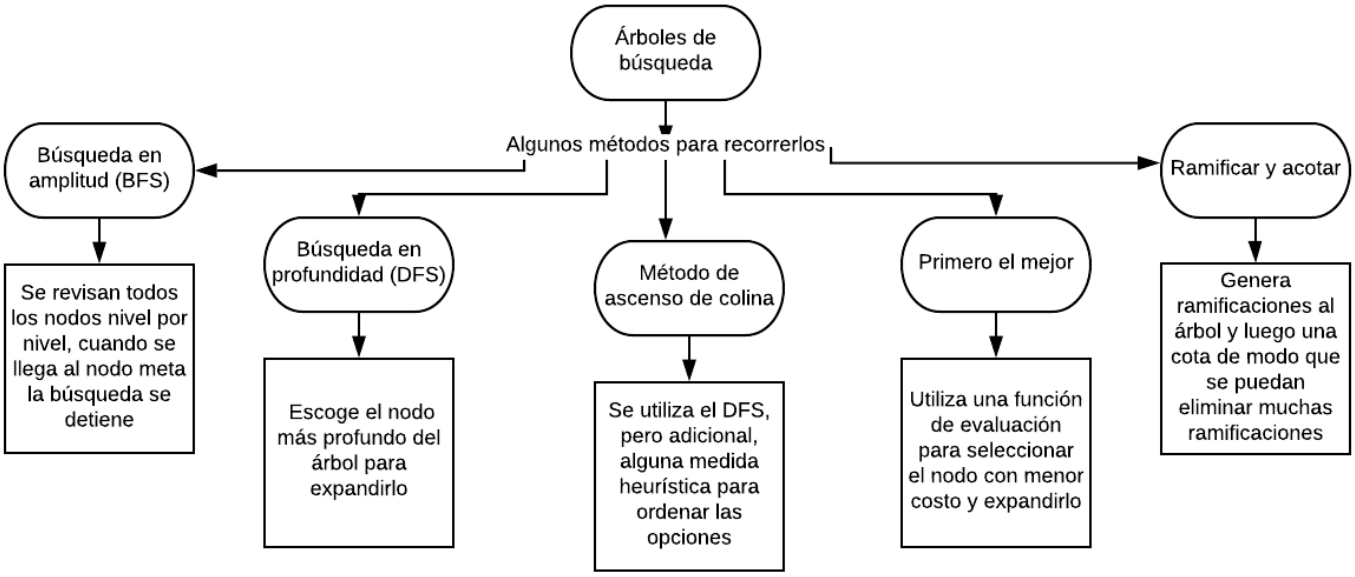
Iniciando en 5: 5

Iniciando en 6: 6, 2, 1, 4, 0, 5, 3, 7

Iniciando en 7: 7

4.5.1 return 1 + lcs(i - 1, j - 1, s1, s2);
4.5.2 return Math.max(ni, nj);
4.5.3 $O(2^n)$
4.6.1 C
4.6.2 A
4.7.1 if (N== r) {
4.7.2 a[r] = i;
4.7.3 if(place(a,r)) sol(a, r+1);

5) Recommended reading (optional)



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

6) Team work and gradual progress

CHANGE REPORT

INTEGRANTE	FECHA	HECHO	HACIENDO	POR HACER
Daniel Felipe Gómez Martinez, Cesar Andres Garcia Posada Y Daniel Garcia Garcia	15/03/2019		Reading and understanding the laboratory.	Methods to work with graphs
Daniel Felipe Gómez Martinez	15/03/2019		Reading the file ".txt" and the Backtracking_Digraph class. (For point 2).	Elaboration of the travel method to create the base or guide to the DFS route.
Cesar Andres Garcia Posada	15/03/2019		Algorithm to find the shortest path between two points using backtracking. (Point 1)	Algorithm N queens Minimum cost algorithm with backtracking
Daniel Garcia Garcia	15/03/2019		Practice for midterm (Point 4)	Practice for midterm (Point 4)
Daniel Felipe Gómez Martinez	16/03/2019	Elaborate the recursive class (For point 2).	Elaborating the findroad method, which recursively performs the search for a path from the successors of a vertex to the final vertex (For point 2).	Method to write the response and comparison method of route comparisons (For point 2).
Cesar Andres Garcia Posada	16/03/2019		Algorithm to find the shortest path between two points using backtracking.	Laboratory Report and test cases. Algorithm N queens

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

				Minimum cost algorithm with backtracking
Cesar Andres Garcia Posada	17/03/2019	Laboratory Report	Algorithm to find the shortest path between two points using backtracking.	Algorithm N queens Minimum cost algorithm with backtracking
Daniel Garcia Garcia	17/03/2019	Practice for midterm (Point 4)	Laboratory report	Laboratory report
Daniel Felipe Gómez Martinez	17/03/2019	Method to write the response and comparison method of route comparisons.	Creating tests for the code.	
Daniel Felipe Gómez Martinez, Cesar Andres Garcia Posada Y Daniel Garcia Garcia	17/03/2019	Laboratory report	Partial simulation questions	Optional points for more learning.

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