

## Laboratorio Nro. 3 Backtracking

**Juan Pablo Rincon Usma**  
Universidad Eafit  
Medellín, Colombia  
jprinconu@eafit.edu.co

**Julián Gómez Benítez**  
Universidad Eafit  
Medellín, Colombia  
jgomezb11@eafit.edu.co

### 3) Simulacro de preguntas de sustentación de Proyectos

**3.1** Aparte de estas dos formas que son las más comunes, se puede implementar algoritmos Greedy tal como Dijkstra, también existe la posibilidad de usar algún algoritmo MST (minimum spanning tree) o algunas técnicas de Dynamic Programming se pueden llegar a implementar.

**3.2** En un grafo dirigido y completo donde  $n$  es el número de nodos, teniendo en cuenta que hay aristas que apuntan de cada nodo a sí mismo, habrían  $n^2$  aristas.

NQueens	Time in ms
4	10
6	12
8	15
10	18
12	20
14	22
16	42
18	61
20	132
22	676
24	315
26	436
28	1765
30	26960
32	62315
<b>3.3</b>	$O(2^n)$

**PhD. Mauricio Toro Bermúdez**  
Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473

## ESTRUCTURA DE DATOS 2

### Código ST0247

**3.4** BFS es mejor utilizarlo cuando se requiere saber la cantidad mínima de pasos que toma ir de un lugar a otro, o generalmente hablando, cuando se quiere determinar la mínima cantidad de pasos que toma ir de un estado inicial a uno final.

DFS se usa más en ocasiones en las cuales uno necesita analizar todas las posibles soluciones para determinar cuál es la mejor.

**3.5** Primero leímos la entrada, y la representamos en una matriz dentro del código, luego implementamos un método que calcula el camino más corto de manera que para el nodo source (donde empieza) se le sacan todos los sucesores y luego se llama recursivamente el método por cada sucesor (se le pasa como source y la destination sigue siendo la misma).

**3.6** La complejidad es  $2^m$ . Teniendo en cuenta que esta es para el peor de los casos y este es imposible que se dé.

**3.7** Siendo m el número aristas en el grafo.

**3.8** Teniendo en cuenta los comentarios de los laboratorios pasados decidimos representar el mapa con listas de adyacencia para mejorar drásticamente el consumo de memoria. Luego usamos un método recursivo (backtracking) para determinar el camino más corto junto con el costo mínimo.

#### 4) Simulacro de Parcial

```

4.1.    1 (n-a, a, b, c)
2 res, solucionar (n-b,a,b,c)+1
3 res, solucionar (n-c,a,b,c)+1
4.2    1 graph.length
2 v, graph, path, pos
3 graph, path, pos +1
4.5    1 1
2 nj, ni
3 T(n) = T(n-3) + c
4.7    1 r >= n
        2 1
3 r+1

```

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
 Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
 Tel: (+57) (4) 261 95 00 Ext. 9473

