ESTRUCTURA DE DATOS 2
Código ST0247

Laboratorio Nro. 3
Vuelta atrás(Backtracking)

Jhonatan Sebastián Acevedo Castrillón
Universidad Eafit
Medellín, Colombia
jsacevedoc@eafit.edu.co

Manuel Alejandro Gutiérrez Mejía
Universidad Eafit
Medellín, Colombia
magutierrm@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1

A*: **El algoritmo A estrella (A star en inglés) es un algoritmo de búsqueda en grafos completos, ya que el algoritmo recibe un nodo inicial y final el cual el algoritmo buscará el camino más corto.**

Dijkstra: **El algoritmo Dijkstra es un algoritmo de búsqueda del camino más corto en grafos creado en 1959, La idea subyacente en este algoritmo consiste en ir explorando todos los caminos más cortos que parten del vértice origen y que llevan a todos los demás vértices; cuando se obtiene el camino más corto desde el vértice origen hasta el resto de los vértices que componen el grafo, el algoritmo se detiene.**

Bellman-Ford: **El algoritmo de Bellman-Ford es menos eficiente que el de Dijkstra pero este puede calcular el camino más corto con pesos negativos, por el contrario Dijkstra no puede.**

3.2 **Habrá n! caminos, donde n es el número de vértices.**

3.3

Backtracking:

Valor de N	Tiempo de ejecución(s)
4	0.001115925
5	0.010963376

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

6	0.002207345
8	0.005411577
10	0.006344878
12	0.013228677
15	0.016089164
25	0.520765315
32	Demora más de 50 minutos.

Fuerza bruta:

Valor de N	Tiempo de ejecución(s)
4	0.003089475
5	0.010063879
6	0.005071281
8	0.224611743
10	1.160370525
12	65.17976913
15	240
25	Demora más de 50 minutos.
32	Demora más de 50 minutos.

3.4 **Es prudente usar BFS cuando queremos encontrar la ruta más corta desde un determinado nodo (origen) a otro nodo (destino), en resumen: El menor número de pasos para alcanzar el estado final desde un estado inicial, en cambio DFS busca todos los posibles caminos agotando todas las posibilidades y al final marca cual es el mejor.**

3.5 **Utilizamos una LinkedList para añadir los caminos que vamos recorriendo para así obtener el más corto, utilizamos esta estructura de datos ya que su método add tiene**

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

complejidad O(1) y su método remove tambien tiene esa misma complejidad, el programa funciona utilizando un for para recorrer todos los sucesores de cada vértices después tenemos un llamado recursivo en el cual iremos recorriendo el árbol, al final cuando vaya hasta el vértice destino, o sea cuando nuestro origen sea igual al tamaño del grafo menos uno, retornará y empezará a crear los otros caminos comparando así cuál es el menor.

3.6 **O(n)*c1**

3.7 **Donde n es el número de vértices y c1 es un parámetro arbitrario.**

3.8 **El algoritmo de la resolución del problema usó un método de actualización de valores que se apoya principalmente en un arreglo cuyas posiciones se irán renovando a partir del costo que represente el arco que parte desde el nodo de inicio hacia sus nodos sucesores, los cuales se iteran por medio de un llamado recursivo hasta llegar al nodo objetivo, asimismo se almacenan y se tienen en cuenta ciertos caminos candidatos por lo que de esta forma se implementó el backtracking.**

4) Simulacro de Parcial

4.1 **Línea 4** n-a,b,c **Línea 5** res,1+ solucionar(n-b,a,b,c)) **Línea 6** res,1+ solucionar(n-c,a,b,c)

4.2 **Línea 2** graph.length **Línea 9** (v,graph,path,pos) **Línea 11** (graph,path,pos+1)

4.3 **DFS** 0,3,7,4,2,1,5,6 **BFS** 0 3 4 2 1 6 5

4.4

4.5 **Línea 7** 1+lcs(i-1,j-1,s1,s2); **Linea 11** return Math.max(ni, nj); **Complejidad:** O(2^n)

4.6 **1)** c **2) b**

4.7 **Línea 3** if(r > N-1) **Línea 8** a[r] = i; **Línea 9** sol(a,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