Reading By Abstraction

Dijkstra

initializeDataStructures

Línea 60: asigna el valor infinito a la posición ‘i’ del array distances.

Línea 61: asigna el valor false a la posición ‘i’ del array visited.

Línea 62: asigna el valor -1 a la posición ‘i’ del array prev.

Líneas 59 - 63: mientras el contador ‘i’ esté dentro del rango del número de vértices (nVertices):

- Asigna el valor infinito a la posición ‘i’ del array distances
- Asigna el valor false a la posición ‘i’ del array visited
- Asigna el valor -1 a la posición ‘i’ del array prev.

Es decir, inicializa los tres arrays con los valores por defecto.

Línea 54: instancia el array distances de tipo Double.

Línea 55: instancia el array visited de tipo Boolean.

Línea 56: instancia el array prev de tipo Integer.

Línea 52 - 66: la función initializeDataStructures() instancia e inicializa los arrays distances, visited y prev.

nextCur

Línea 79: le da el valor de i a la variable next.

Línea 80: cost pasa a tener el valor de la posición i del array distances.

Línea 78-81: si el nodo i no ha sido visitado y la distancia de i es menor que cost, la variable next pasa a valer i y la variable cost pasa a valer la distancia de i.

Línea 74: se define la variable next como un int que vale -1 inicialmente.

Línea 75: se define la variable cost como un double que pasa a valer infinito con la sentencia 'Double.POSITIVE_INFINITY'.

Línea 77-82: mientras, a partir de 0, i sea menor que número de vértices:

- si el nodo i no ha sido visitado y la distancia de i es menor que cost, la variable next pasa a valer i y la variable cost pasa a valer la distancia de i.

Línea 73-85: el método nextCur: inicializa variables y mediante un bucle for recorre los vértices para devolver el nodo que no haya sido visitado y que menor coste tenga.

getPath

Línea 156: La variable error pasa a ser true.

Línea 157: se devuelve null;

Línea 155-158: si la ejecución de dijkstra no se ha cumplido(!dijkstraExec), a la variable error se le asigna true, y se devuelve null.

Línea 160: se define un ArrayList de enteros llamado path.

Línea 164: se define un entero llamado cur, y se inicializa a end que es el último valor del algoritmo (variable pasada por parámetro).

Línea 165: se añade el valor de la variable cur al ArrayList path.

Línea 167: se añade al ArrayList path el nodo previo a cur.

Línea 168: El valor de cur pasa a tener el valor de su nodo anterior.

Línea 166-169: mientras el valor de cur no sea el primer valor del algoritmo (ini),

- se añade al ArrayList path el nodo previo a cur.
- El valor de cur pasa a tener el valor de su nodo anterior.

Línea 153-172: el método getPath inicializa variables, y devuelve un array que contiene el camino más corto desde el vértice inicial del algoritmo, hasta el último.

hasErrorHappened

Línea 181: se devuelve el valor de la variable error.

Línea 180-182: el método hasErrorHappened devuelve el valor de la variable error, que inicialmente vale false, pero si ocurre algún error al ejecutar valdrá true.

computeShortestPath

Línea 126: asigna al contenido del índice ‘neigh’ del array distances el valor de ‘newDistance’

Línea 127: asigna al contenido del índice ‘neigh’ del array prev el valor de cur.

Línea 122: asigna a la variable ‘newDistance’ la suma del valor del índice ‘cur’ del array distances con el valor de los índices ‘cur’ y ‘neigh’ de la matriz adjMatrix.

Línea 125: Si el valor de la posición ‘neigh’ del array distances es estrictamente mayor que ‘newDistance’:

- Asigna al contenido del índice ‘neigh’ del array prev el valor de cur
- Asigna a la variable ‘newDistance’ la suma del valor del índice ‘cur’ del array distances con el valor de los índices ‘cur’ y ‘neigh’ de la matriz adjMatrix

Línea 120: Si el valor de los índices ‘cur’ y ‘neigh’ de la matriz adjMatrix es true y el valor del índice ‘neigh’ del array es false:

- Asigna a la variable ‘newDistance’ la suma del valor del índice ‘cur’ del array distances con el valor de los índices ‘cur’ y ‘neigh’ de la matriz adjMatrix.
- Si el valor de la posición ‘neigh’ del array distances es estrictamente mayor que ‘newDistance’:
    - Asigna al contenido del índice ‘neigh’ del array prev el valor de cur
    - Asigna a la variable ‘newDistance’ la suma del valor del índice ‘cur’ del array distances con el valor de los índices ‘cur’ y ‘neigh’ de la matriz adjMatrix

---

Línea 117: mientras el contador ‘neigh’ esté dentro del rango del número de vértices (nVertices):

- Si el valor de los índices ‘cur’ y ‘neigh’ de la matriz adjMatrix es true y el valor del índice ‘neigh’ del array es false:
    - Asigna a la variable ‘newDistance’ la suma del valor del índice ‘cur’ del array distances con el valor de los índices ‘cur’ y ‘neigh’ de la matriz adjMatrix.
    - Si el valor de la posición ‘neigh’ del array distances es estrictamente mayor que ‘newDistance’:

- Asigna al contenido del índice ‘neigh’ del array prev el valor de cur
- Asigna a la variable ‘newDistance’ la suma del valor del índice ‘cur’ del array distances con el valor de los índices ‘cur’ y ‘neigh’ de la matriz adjMatrix

Línea 132: asigna el valor true a la posición ‘cur’ del array visited

Línea 133: asigna a la variable ‘cur’ el valor devuelto por la función “nextCur”.

Línea 115: Mientras que el valor del índice ‘end’ del array visited sea false:

- Mientras el contador ‘neigh’ esté dentro del rango del número de vértices (nVertices):
    - Si el valor de los índices ‘cur’ y ‘neigh’ de la matriz adjMatrix es true y el valor del índice ‘neigh’ del array es false:
        - Asigna a la variable ‘newDistance’ la suma del valor del índice ‘cur’ del array distances con el valor de los índices ‘cur’ y ‘neigh’ de la matriz adjMatrix.
        - Si el valor de la posición ‘neigh’ del array distances es estrictamente mayor que ‘newDistance’:
            - Asigna al contenido del índice ‘neigh’ del array prev el valor de cur
            - Asigna a la variable ‘newDistance’ la suma del valor del índice ‘cur’ del array distances con el valor de los índices ‘cur’ y ‘neigh’ de la matriz adjMatrix
- Asigna el valor true a la posición ‘cur’ del array visited
- Asigna a la variable ‘cur’ el valor devuelto por la función “nextCur”.

Línea 110: se define un Double llamado newDistamce.

Línea 111: se llama a la función initializeDataStructures().

Línea 113: se asigna el entero ini a la variable cur.

Línea 114: se inicializa el array distances en la posición ini al valor 0.0

Línea 136: se asigna true a la variable dijkstraExec.

Línea 137: se devuelve el array distances en la posición end.

Análisis de nextCur

	Código	Lo que se supone que hace	Lo que creemos que hace
2	private Integer nextCur() {	Returns the next vertex to be analyzed by the Dijkstra algorithm. As required, that should be the unvisited vertex with smallest expected cost. Return Next vertex to be explored.	Cada vez que se llama a nextCur, tras inicializar next a -1 y coste a un valor alto, se entra en un for que recorre todos los nodos del grafo hasta encontrar el nodo no visitado a menor distancia. Por último, retorna ese nodo.
3			
4	Integer next = -1;		
5			
6	Double cost = Double.POSITIVE_INFINITY;		
7			
8	for(int i=0; i<nVertices; i++) {		
9			
10	if(!visited[i] && distances[i]<cost) {		
11			
12	next = i;		
13			
14	cost = distances[i];		
15			
16	}		
17			
18	}		
19			
20	return next;		
21			
22	}		