JavaScript is disabled on your browser.

[Skip navigation links](#1fob9te)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/GraphAlgorithmObserver.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* Prev Class
* [Next Class](http://docs.google.com/graph/MazeGraph.html)
* [Frames](http://docs.google.com/index.html?graph/GraphAlgorithmObserver.html)
* [No Frames](http://docs.google.com/GraphAlgorithmObserver.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* Constr |
* [Method](#3znysh7)
* Detail:
* Field |
* Constr |
* [Method](#2et92p0)

graph

## Interface GraphAlgorithmObserver<V>

* Type Parameters: V -  
    
  public interface GraphAlgorithmObserver<V>  
  A GraphAlgorithmObserver will register with a WeightedGraph to let it know that it is "watching". As algorithms are carried out on the WeightedGraph (BFS, DFS, Dijkstra), the graph will notify the Observer to let it know how the algorithms are progressing.Author: Fawzi Emad (C) 2016

### Method SummaryAll Methods Instance Methods Abstract Methods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + void | * + [notifyBFSHasBegun](http://docs.google.com/graph/GraphAlgorithmObserver.html#notifyBFSHasBegun--)() Called by the graph to notify this Observer that a Breadth-First Search has been initiated. |
| * + void | * + [notifyDFSHasBegun](http://docs.google.com/graph/GraphAlgorithmObserver.html#notifyDFSHasBegun--)() Called by the graph to notify this Observer that a Depth-First-Search has been initiated. |
| * + void | * + [notifyDijkstraHasBegun](http://docs.google.com/graph/GraphAlgorithmObserver.html#notifyDijkstraHasBegun--)() Called by the graph to notify this observer that Dijkstra's algorithm has begun. |
| * + void | * + [notifyDijkstraIsOver](http://docs.google.com/graph/GraphAlgorithmObserver.html#notifyDijkstraIsOver-java.util.List-)(java.util.List<[V](http://docs.google.com/graph/GraphAlgorithmObserver.html)> path) Called by the graph to notify this observer that Dijkstra's algorithm is over. |
| * + void | * + [notifyDijkstraVertexFinished](http://docs.google.com/graph/GraphAlgorithmObserver.html#notifyDijkstraVertexFinished-V-java.lang.Integer-)([V](http://docs.google.com/graph/GraphAlgorithmObserver.html) vertexAddedToFinishedSet, java.lang.Integer costOfPath) Called by the graph to notify this observer that a vertex has been added to the "Finished Set" during Dijkstra's algorithm. |
| * + void | * + [notifySearchIsOver](http://docs.google.com/graph/GraphAlgorithmObserver.html#notifySearchIsOver--)() Called by the graph to notify this observer that the search (either DFS or BFS) is over. |
| * + void | * + [notifyVisit](http://docs.google.com/graph/GraphAlgorithmObserver.html#notifyVisit-V-)([V](http://docs.google.com/graph/GraphAlgorithmObserver.html) vertexBeingVisited) Called by the graph to notify this Observer that a vertex is being "visited" during either DFS or BFS. |

### Method Detail

#### notifyDFSHasBegun void notifyDFSHasBegun() Called by the graph to notify this Observer that a Depth-First-Search has been initiated.

#### notifyBFSHasBegun void notifyBFSHasBegun() Called by the graph to notify this Observer that a Breadth-First Search has been initiated.

#### notifyVisit void notifyVisit([V](http://docs.google.com/graph/GraphAlgorithmObserver.html) vertexBeingVisited) Called by the graph to notify this Observer that a vertex is being "visited" during either DFS or BFS.Parameters: vertexBeingVisited -

#### notifySearchIsOver void notifySearchIsOver() Called by the graph to notify this observer that the search (either DFS or BFS) is over.

#### notifyDijkstraHasBegun void notifyDijkstraHasBegun() Called by the graph to notify this observer that Dijkstra's algorithm has begun.

#### notifyDijkstraVertexFinished void notifyDijkstraVertexFinished([V](http://docs.google.com/graph/GraphAlgorithmObserver.html) vertexAddedToFinishedSet, java.lang.Integer costOfPath) Called by the graph to notify this observer that a vertex has been added to the "Finished Set" during Dijkstra's algorithm. The second parameter is the "cost" (total weight) of the best path leading from the starting vertex to the one referenced by the first parameter.Parameters: vertexAddedToFinishedSet - costOfPath -

#### notifyDijkstraIsOver void notifyDijkstraIsOver(java.util.List<[V](http://docs.google.com/graph/GraphAlgorithmObserver.html)> path) Called by the graph to notify this observer that Dijkstra's algorithm is over.Parameters: path - A list of Vertices that are connected along edges, beginning with the "starting vertex" and ending with the "finishing vertex". This will be the optimal (lowest cost) path from start to finish.

[Skip navigation links](#44sinio)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/GraphAlgorithmObserver.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* Prev Class
* [Next Class](http://docs.google.com/graph/MazeGraph.html)
* [Frames](http://docs.google.com/index.html?graph/GraphAlgorithmObserver.html)
* [No Frames](http://docs.google.com/GraphAlgorithmObserver.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* Constr |
* [Method](#3znysh7)
* Detail:
* Field |
* Constr |
* [Method](#2et92p0)

[Web Accessibility](https://www.umd.edu/web-accessibility)