Diseño de pruebas unitarias

Alejandro Arce

Duvan Cuero

Alexander Samaca

Universidad ICESI

Algoritmos y Estructuras de datos

Santiago de Cali

Noviembre 2020

**Diseños de pruebas unitarias**

En este documento se desarrollará la documentación de los casos de pruebas del mundo del problema, así como de las estructuras de datos

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpEmptyGraph | GraphTest | An empty Graph. |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpGraph | GraphTest | A graph with 5 vertex  With values 1,2,3,4,5  [1,2]  [2,3]  [2,4]  [4,5]  [1,5] |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpGraph1 | GraphTest | A graph with 5 vertex  With values 1,2,3,4,5  x, y, w [1, 2, 2] [2, 1, 2] [1, 3, 12] [3, 1, 12] [2, 3, 7] [3, 2, 7] [2, 4, 15] [4, 2, 15] [3, 4, 3] [4, 3, 3] [2, 5, 4] [5, 2, 4] [4, 5, 6] [5, 4, 6] |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpGraph2 | GraphTest | A graph with 4 vertex  With values 1,2,3,4  x, y, w [1, 2, 1] [2, 3, 2] [1, 4, 5] [3, 4, 1] |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpGraph3 | GraphTest | A graph with 5 vertex  With values 1,2,3,4,5  x, y, w [1, 2, 5] [1, 3, 1] [1, 4, 3] [2, 5, 8] [4, 2, 2] [4, 5, 5] [4, 3, 2] |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** addVertex, comprobar que el método addVertex añade correctamente un vértice al grafo | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| GraphTest | AddVertex | setUpEmpyGraph | value = 1 | El resultado esperado es que el método añada un vértice al grafo |
| GraphTest | AddVertex | setUpGraph | value=7  value=8 | El resultado esperado es que el método añada un vértice al grafo |
| GraphTest | AddVertex | setUpGraph | value=2 | El resultado esperado es que el método no añada un vértice ya que existe un vértice con ese valor |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** removeVertex, comprobar que el método removeVertex elimine correctamente un vértice del grafo | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| GraphTest | removeVertex | setUtGraph | value = 1 | El resultado esperado es que el método elimine un vértice del grafo |
| GraphTest | removeVertex | setUtEmptyGraph | value = 1 | El resultado esperado es que el método no elimine un vértice del grafo ya que no existe vértice con ese valor |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** addEdge, comprobar que el método addEdge añade correctamente una arista al grafo | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| GraphTest | AddEdge | setUpEmpyGraph | From=1  To=2 | El resultado esperado es que el método no añada una arista ya que no hay vértices |
| GraphTest | AddEdge | setUpGraph | From=1  To=5 | El resultado esperado es que el método añada una arista entre los vértices 1 y 5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** removeEdge, comprobar que el método removeEdge elimine correctamente una arista del grafo | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| GraphTest | removeEdge | setUpEmpyGraph | From=1  To=2 | El resultado esperado es que el método no elimine una arista ya que no hay aristas |
| GraphTest | removeEdge | setUpGraph | From=1  To=2 | El resultado esperado es que el método elimine una arista que va de 1 a 2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** dfs, comprobar que el método dfs hace correctamente el recorrido por el grafo | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| GraphTest | dfs | setUpEmpyGraph |  | El resultado esperado es que el método no realice ningún recorrido ya que no hay vértices. |
| GraphTest | dfs | setUpGraph |  | El resultado esperado es que el método realice recorrido correctamente con una salida de 1,2,3,4,5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** bfs, comprobar que el método bfs hace correctamente el recorrido por el grafo | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| GraphTest | bfs | setUpEmpyGraph |  | El resultado esperado es que el método no realice ningún recorrido ya que no hay vértices. |
| GraphTest | bfs | setUpGraph |  | El resultado esperado es que el método realice recorrido correctamente con una salida de 1,2,5,3,4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** prim, comprobar que el método prim asigna los valores de los vértices correctamente formando un árbol de expansión minima | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| GraphTest | prim | setUpEmpyGraph |  | El resultado esperado es que el método no asigne ningún valor ya que no hay vértices. |
| GraphTest | prim | setUpGraph1 |  | El resultado esperado es que el método asigne los valores correctamente con una salida de nil,1,4,5,2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** krustal, comprobar que el método krustal asigna los valores de los vértices correctamente formando un árbol de expansión mínima | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| GraphTest | krustal | setUpEmpyGraph |  | El resultado esperado es que el método no asigne ningún valor ya que no hay vértices. |
| GraphTest | krustal | setUpGraph1 |  | El resultado esperado es que el método forme correctamente las aristas para hacer una árbol de expansión mínima con una salida de { (1,2), (3,4), (2,5), (4,5) } |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** floyd-Warshall comprobar que el método floyd-Warshall encuentra la mínima distancia entre un vértice y resto de vértices del grafo | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| GraphTest | Floyd-Warshall | setUpEmpyGraph |  | El resultado esperado es que el método no encuentre ninguna mínima distancia ya que no hay vértices. |
| GraphTest | Floyd-Warshall | setUpGraph2 |  | El resultado esperado es que el método retorne la matriz con las mínimas distancias  1 2 3 4  [ 0 1 3 4]  [ ∞ 0 2 3]  [ ∞ ∞ 0 1]  [ ∞ ∞ ∞ 0] |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** dijkstra comprobar que el método dijkstra encuentra la mínima distancia entre un vértice y resto de vértices del grafo | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| GraphTest | dijkstra | setUpEmpyGraph | Vertex source = 1  Vertex dest = 5 | El resultado esperado es que el método no encuentre ninguna mínima distancia ya que no hay vértices. |
| GraphTest | dijkstra | setUpGraph2 | Vertex source = 1  Vertex dest = 5 | El resultado esperado es que el método retorne los vértices que debe recorrer con menor distancia 1,4,5 |