Sebastian Escobar Marin – A00374994

Juan Esteban Duque Taborda - A00376778

|  |  |
| --- | --- |
| **Escenario de caso de prueba** | |
| **Identificador** | setupStage1 |
| **graph** | Cuenta de usuario activa |
| **Variables** | **Estado** |
| 1. **Tree** | Se inicializa un objeto graph de representado como lista de vecinos. |

|  |  |
| --- | --- |
| **Escenario de caso de prueba** | |
| **Identificador** | setupStage2 |
| **graph** | Cuenta de usuario activa |
| **Variables** | **Estado** |
| 1. **Tree** | Se inicializa un objeto graph de representado como matriz de adyacencia. |

|  |  |
| --- | --- |
| **Formato caso de prueba** | |
| **Escenario** | setupStage1 |
| **Identificador** | CP-01 |
| **Nombre del caso** | Prueba Dijkstra grafo de lista de vecinos |
| **Precondiciones** | Árbol inicializado |
| **Paso** | **Resultado esperado** |
| 1. Se añaden vértices al grafo. 2. Se añaden dos aristas una de “a” a “b” de peso 5 y otra de “b” a “c” de peso 1. 3. Se llama el método path desde “a” hasta “c”. | La lista de nodos tiene un tamaño de 3.  La lista de aristas de el vértice “a” y “b” debe tener tamaño de 1.  El vértice previo de “c” es “b” y el de “b” es “a”, asimismo la variable distancia del vértice “c” es igual a 6. |

|  |  |
| --- | --- |
| **Formato caso de prueba** | |
| **Escenario** | setupStage2 |
| **Identificador** | CP-02 |
| **Nombre del caso** | Prueba Floyd Warshall grafo de matriz de adyacencia |
| **Precondiciones** | Árbol inicializado |
| **Paso** | **Resultado esperado** |
| 1. Se añaden vértices al grafo. 2. Se añaden dos aristas una de “a” a “b” de peso 5 y otra de “b” a “c” de peso 1. 3. Se llama el método path desde “a” hasta “c”. | La matriz de nodos debe tener tamaño 3x3.  La entrada a,b de la matriz es igual a 5 y la entrada b,c es igual a 1.  En la matriz de antecesores la entrada a,c es b y la entrada a,b es a. En la matriz solución del Floyd Warshall la entrada a,c es igual a 6. |

|  |  |
| --- | --- |
| **Formato caso de prueba** | |
| **Escenario** | setupStage1 |
| **Identificador** | CP-03 |
| **Nombre del caso** | Prueba añadir y eliminar vértices y aristas grafo de lista de vecinos |
| **Precondiciones** | Árbol inicializado |
| **Paso** | **Resultado esperado** |
| 1. Se añaden vértices al grafo “a”, “b” y “c”. 2. Se añade una arista de “a” a “b” de peso 10. 3. Se elimina una arista de “a” a “b” de peso 10. 4. Se elimina el vértice “c”. | La lista de nodos tiene un tamaño de 3.  La lista de aristas de el vértice “a” debe tener tamaño de 1.  La lista de aristas de el vértice “a” debe tener tamaño de 0.  La lista de nodos tiene un tamaño de 2. |

|  |  |
| --- | --- |
| **Formato caso de prueba** | |
| **Escenario** | setupStage2 |
| **Identificador** | CP-04 |
| **Nombre del caso** | Prueba añadir y eliminar vértices y aristas grafo de matriz de adyacencia |
| **Precondiciones** | Árbol inicializado |
| **Paso** | **Resultado esperado** |
| 1. Se añaden vértices al grafo “a”, “b” y “c”. 2. Se añade una arista de “a” a “b” de peso 10. 3. Se elimina una arista de “a” a “b” de peso 10. 4. Se elimina el vértice “c”. | La matriz de nodos debe tener tamaño 3x3.  La entrada a,b de la matriz es igual a 10.  La entrada a,b de la matriz es igual a el máximo valor de un entero en Java.  La matriz de nodos debe tener tamaño 2x2. |