

Pruebas Grafos

Prueba No. 1	Objetivo de la prueba: Mostrar que funcionan los principales métodos de las aristas del grafo. Firma del método: public void getWeight () y setWeight(int)			
Clase para probar	Método	Escenario	Valores de entrada	Resultado
Edge	getWeight()	setupStage1, se crea un grafo y se añaden nodos y aristas con valores:		
Edge	setWeight(int)	setupStage1, se crea un grafo y se añaden nodos y aristas con valores:		

Pruebas Grafo por Lista de Adyacencia

Prueba No. 1	Objetivo de la prueba: Mostrar que se agregan elementos al árbol Firma del método: public void (t)			
Clase para probar	Método	Escenario	Valores de entrada	Resultado
GraphAL	insert(T)	setupStage2, se crea un grafoAL	3	Arbol ABB j= vacío

Prueba No. 2	Objetivo de la prueba: Mostrar que se encuentran los valores buscando en el árbol Firma del método: public Node<T> find(t)			
Clase para probar	Método	Escenario	Valores de entrada	Resultado
GraphAL	delete(T)	setupStage2, se crea un grafoAL		

Prueba No. 3	Objetivo de la prueba: Mostrar que se elimina un elemento del árbol Firma del método: public void delete(T)			
Clase para probar	Método	Escenario	Valores de entrada	Resultado
NodeGraphAL	delete(T)	setupStage2	7	Find(7) == null

Prueba No. 4	Objetivo de la prueba: Mostrar que dado un nodo se obtiene su sucesor Firma del método: public Node<T> getSucesor(T)			
Clase para probar	Método	Escenario	Valores de entrada	Resultado
NodeGraphAL	getSucesor(T)	setupStage2	5	6

Pruebas Grafo por Matriz de Adyacencia

Prueba No. 1	Objetivo de la prueba: Mostrar que se agregan elementos al árbol y este queda aproximadamente balanceado haciendo las debidas rotaciones y recoloraciones Firma del método: public void insert(t)			
Clase para probar	Método	Escenario	Valores de entrada	Resultado
GraphAM	insert(T)	setupStage1, se crea un Arbol RBT y se inserta los nodos con valores 5, 4, 6, 3, 7	8	Find(7).color == BLACK && find(7).left == 6 && find(7).right == 8

Prueba No. 2	Objetivo de la prueba: Mostrar que se encuentran los valores buscando en el árbol Firma del método: public Node<T> find(t)			
Clase para probar	Método	Escenario	Valores de entrada	Resultado

GraphAM	find(T)	setupStage1	7	7
---------	---------	-------------	---	---

Prueba No. 3	Objetivo de la prueba: Mostrar que el árbol queda aproximadamente balanceado luego de eliminar un nodo negro, haciendo las debidas rotaciones y recoloraciones Firma del método: public void delete(T)			
Clase para probar	Método	Escenario	Valores de entrada	Resultado
NodeGraphAM	delete(T)	setupStage2, se crea un Arbol RBT y se inserta los nodos con valores 5, 4, 6, 3, 7, 8	7	Find(7) == null, root.left == 4 && root.left == 8 && find(8).color == BLACK

Prueba No. 4	Objetivo de la prueba: Mostrar que dado un valor se obtiene su sucesor Firma del método: public Node<T> getSucesor(T)			
Clase para probar	Método	Escenario	Valores de entrada	Resultado
NodeGraphAM	getSucessor(T)	setupStage2	5	6