**Escenarios de prueba**

**MaxColaPrioridad:**

***Escenario 1.* Adición de un elemento de prioridad máxima a un Queue de prioridad creado. Descripción:** Se busca corroborar que la clase MaxColaPrioridad agregue correctamente un nuevo elemento a un Queue ordenado máximo para prioridad. Esta adición será corroborada con un Queue de un tamaño superior a 4 para asegurar una correcta reorganización de este.

***Escenario 2.* Adición de un elemento de prioridad mínima a un Queue de prioridad creado. Descripción:** En este escenario se verificará que el elemento de prioridad mínima sea agregado al final del Queue, de tal manera que este quede organizado al final de la iteración. Para este propósito se utilizará un Queue de tamaño superior a 4 elementos, de tal manera que exista un número considerable de elementos en la implementación.

***Escenario 3.* Adición de un elemento de prioridad intermedia a un Queue de prioridad creado. Descripción:** Se busca observar el organizamiento de un Queue para un elemento adicional que no va a ser añadido en uno de los extremos de este, la reorganización de cada elemento ca a ser estudiada con este escenario; observando el número de recorridos y movimientos, así como la posición final del nuevo elemento se definirá que óptimo es el algoritmo.

***Escenario 4.* Adición de un elemento a un Queue de prioridad vacío. Descripción:** Con este escenario se busca estudiar el comportamiento de Queue para la adición de un nuevo elemento en su primera etapa. De no añadirse el nuevo elemento se encuentra un error de algoritmo en el método agregar.

**MaxHeapCP**

***Escenario 1*. Adición de un elemento de prioridad máxima a un Heap de prioridad creado. Descripción:** Se busca corroborar que la clase MaxHeapCP agregue correctamente un nuevo elemento a un Queue ordenado máximo para prioridad. Esta adición será corroborada con un Queue de un tamaño superior a 4 para asegurar una correcta reorganización de este.

***Escenario 2.* Adición de un elemento de prioridad mínima a un Heap de prioridad creado. Descripción:** En este escenario se verificará que el elemento de prioridad mínima sea agregado al final del Heap, de tal manera que este quede organizado al final de la iteración. Para este propósito se utilizará un Heap de tamaño superior a 4 elementos, de tal manera que exista un número considerable de elementos en la implementación.

***Escenario 3.* Adición de un elemento de prioridad intermedia a un Heap de prioridad creado. Descripción:** Se busca observar el organizamiento de un Heap para un elemento adicional que no va a ser añadido en uno de los extremos de este, la reorganización de cada elemento ca a ser estudiada con este escenario; observando el número de recorridos y movimientos, así como la posición final del nuevo elemento se definirá que óptimo es el algoritmo.

***Escenario 4.* Adición de un elemento a un Heap de prioridad vacío. Descripción:** Con este escenario se busca estudiar el comportamiento de Heap para la adición de un nuevo elemento en su primera etapa. De no añadirse el nuevo elemento se encuentra un error de algoritmo en el método agregar.

GRAFICA DE TIEMPO PROMEDIO DE EJECUCIÓN V.S. NÚMERO DE DATOS.

Agregar:

|  |  |
| --- | --- |
| N | Tiempo(mseg) |
| 50000 | 170358 |
| 100000 | 719141 |
| 150000 | 2062483 |
| 200000 | 4545385 |
| 250000 | 7393401 |
| 300000 | 11314725 |