Análisis de eficiencia

|  |  |  |
| --- | --- | --- |
|  | MaxColaCP | MaxHeapCP |
| ¿Cuándo ocurre el peor caso de agregar()? | Se presenta cuando se debe el último elemento de la cola y no existen referencias, por lo que se debe mirar todo el arreglo. | Se presenta cuando el elemento agregado es el mayor en el arreglo, ya que se deben hacer Log(N) intercambios para llegar hasta la raíz. |
| Complejidad peor caso de agregar() | O(N) | O(Log(N)) |
| ¿Cuándo ocurre el mejor caso de agregar? | Se presenta cuando existe una referencia al último elemento de la cola y solo se debe agregar después de este. | Se presenta cuando el elemento agregado es el menor del arreglo, por lo que se agrega al final del arreglo. |
| Complejidad mejor caso de agregar | O(1) | O(1) |
| Tiempo promedio de agregar() (Promedio de agregar 20,000 datos a una cola vacía) | 14.1 | 25.1 |
| ¿Cuándo ocurre el peor caso de sacarMax()? | Aunque exista una referencia al elemento mayor de la cola, se debe buscar en toda la lista para asignar el nuevo elemento mayor. | Aunque el elemento máximo siempre esta en la posición 1, la raíz, se debe buscar el nuevo elemento que ocupara la posición de max. En el peor caso, el nuevo elemento máximo está al final del heap. |
| Complejidad peor caso de sacarMax() | O(N) | O(2\*Log(N)) |
| ¿Cuándo ocurre el mejor caso de sacarMax()? | Ya que la estructura está desordenada, se debe mirar todo el arreglo para saber cuál es el elemento mayor y para asignar el nuevo elemento mayor. | Aunque el elemento máximo siempre esta en la posición 1, la raíz, se debe buscar el nuevo elemento que ocupara la posición de max. En el mejor caso, el nuevo elemento máximo está en la posición 2. |
| Complejidad mejor caso de sacarMax() | O(N) | O(Log(N)) |
| Tiempo promedio de sacarMax() (Promedio de sacar 20,000 datos de una cola con 20,000 datos) | 1200000 | 14.5 |