|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File | ADT# | Time#1 | Time#2 | Time#3 |
| File1 | 1 | 0.107086 | 0.108801 | 0.108422 |
| File2 | 1 | 57.5857 | 57.5804 | 57.5366 |
| File3 | 1 | 0.084307 | 0.083150 | 0.082953 |
| File4 | 1 | 32.8634 | 32.5877 | 32.631 |
| File1 | 2 | 0.092807 | 0.088998 | 0.089057 |
| File2 | 2 | 300.682 | 300.524 | 301.347 |
| File3 | 2 | 0.109212 | 0.105518 | 0.105533 |
| File4 | 2 | 149.947 | 155.752 | 170.386 |
| File1 | 3 | 0.068985 | 0.068959 | 0.069078 |
| File2 | 3 | 0.057618 | 0.057664 | 0.057597 |
| File3 | 3 | 0.057967 | 0.057752 | 0.057881 |
| File4 | 3 | 0.061039 | 0.060822 | 0.060731 |
| File1 | 4 | 0.083801 | 0.083849 | 0.083534 |
| File2 | 4 | 0.069156 | 0.07025 | 0.070618 |
| File3 | 4 | 0.069105 | 0.069781 | 0.069841 |
| File4 | 4 | 0.072085 | 0.072909 | 0.072931 |
| File1 | 5 | 0.074723 | 0.074396 | 0.074214 |
| File2 | 5 | 0.061192 | 0.060715 | 0.06091 |
| File3 | 5 | 0.061934 | 0.061755 | 0.061743 |
| File4 | 5 | 0.065572 | 0.065014 | 0.065093 |
| File1 | 6 | 0.304766 | 0.311911 | 0.309608 |
| File2 | 6 | 0.228541 | 0.229702 | 0.232232 |
| File3 | 6 | 0.265299 | 0.270016 | 0.274708 |
| File4 | 6 | 0.463072 | 0.490939 | 0.494689 |

Big O:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| File | ADT# | Insert | Delete | Entire series of Inserts | Series of Deletions | Entire File |
| 1(In) | 1(LinkedList) | O(1) | O(0) | N | 0 | N |
| 2(In/Del) | 1 | O(1) | O(n) | N | N2 | N2 |
| 3(In/Delrev) | 1 | O(1) | O(1) | N | N | N |
| 4(rand) | 1 | O(1) | O(n) | N | N2 | N2 |
| 1 | 2(CursorList) | O(1) | O(N) | N | 0 | N |
| 2 | 2 | O(1) | O(N) | N | N2 | N2 |
| 3 | 2 | O(1) | O(N) | N | N | N |
| 4 | 2 | O(1) | O(N) | N | N2 | N2 |
| 1 | 3(StackAr) | O(1) | O(1) | N | N | N |
| 2 | 3 | O(1) | O(1) | N | N | N |
| 3 | 3 | O(1) | O(1) | N | N | N |
| 4 | 3 | O(1) | O(1) | N | N | N |
| 1 | 4(StackLi) | O(1) | O(1) | N | N | N |
| 2 | 4 | O(1) | O(1) | N | N | N |
| 3 | 4 | O(1) | O(1) | N | N | N |
| 4 | 4 | O(1) | O(1) | N | N | N |
| 1 | 5(QueueAr) | O(1) | O(1) | N | N | N |
| 2 | 5 | O(1) | O(1) | N | N | N |
| 3 | 5 | O(1) | O(1) | N | N | N |
| 4 | 5 | O(1) | O(1) | N | N | N |
| 1 | 6(SkipList) | O(log n) | O(log n) | N log n | N log n | N log n |
| 2 | 6 | O(log n) | O(log n) | N log n | N log n | N log n |
| 3 | 6 | O(log n) | O(log n) | N log n | N log n | N log n |
| 4 | 6 | O(log n) | O(log n) | N log n | N log n | N log n |

Essay:

LinkedList vs. LinkedList (different test files)

Linked Lists are typically unsorted since they have no method with which they can utilize binary search. For this reason, one may insert easily at the beginning of a linked list without worry. Also, it takes constant (O(1)) time to insert at the beginning of a list since one does not have to push everything back one node. Luckily, this makes the linked list a very efficient data structure for test 1, which is all inserts.

CursorList vs. CursorList (different test files)

StackAr vs. StackAr (different test files)

StackLi vs. StackLi (different test files)

QueueAr vs. QueueAr (different test files)

SkipList vs SkipList (different test files)