|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data Structures | Access | | | Insert | | | Remove | | | Search | | | When to use |
| Best | Average | Worst | Best | Average | Worst | Best | Average | Worst | Best | Average | Worst |
| ArrayList | O(1) | | | At end: O(1) | O(n) | At start: O(n) | From end: O(1) | O(n) | From start: O(n) | Found at start: O(1) | O(n) | Found at end: O(n) |  |
| SLL\* | O(1) | O(n) | O(n) | At start: O(1) | O(n) | At end: O(n) | From start: O(1) | O(n) | From end  O(n) | Found at start: O(1) | O(n) | Found at end: O(n) |  |
| DLL\* | O(1) | O(n) | O(n) | At start/end: O(1) | O(n) | O(n) (avg) | From start/ end: O(1) | O(n) | O(n) (avg) | Found at start: O(1) | O(n) | Found at end: O(n) |  |
| Queue \*\* |  | | | O(1) | | | O(1) | | |  | | |  |
| Binary Tree |  | | |  | | |  | | | O(1) | O(n) | O(n) |  |
| Stack |  | | | O(1) | | | O(1) | | |  | | |  |
| BST |  |  |  |  |  |  |  |  |  |  |  |  |  |

\*: Insertion and deletion at a specified position. The task of insertion/deletion itself takes O(1) time, but accessing that position in the list takes O(n) time

\*\*: insert to back, remove from front

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |