|  |  |  |
| --- | --- | --- |
| Linked List Implementation Time Complexities | | |
| Function Name | **Time Complexity** | **Reason** |
| add\_front( val ) | O(1) | Insertion only requires updating the pointer in the beginning. |
| add\_end( val ) | O(n) | Insertion requires traversing to the end and updating the pointer in the end. |
| peek() | O(1) | The head node is accessed directly. |
| peek\_end() | O(n) | This requires traversing through the entire list and then accessing the last node (tail). |
| get\_size() | O(1) | This is a pretty straightforward function that needs to access the size variable directly |
| print\_list() | O(n) | There is a while loop iterating through the entire list. |
| clear() | O(1) | No complex processes, only clearing the entire list. |
| Is\_empty() | O(1) | Checks the if size = 0. |
| remove\_end() | O(n) | Iterates through the entire list and removes the end by updating the second last pointer to none. |
| remove\_position(position) | O(n) | Iterates through the entire list and finds the element with the position and updates the pointer before it. |
| remove\_beginning() | O(1) | Just updates the head pointer and decrements size. |
| \_make\_list() | O(n) | Traverses through the entire linked list to make a Python list. |

|  |  |  |
| --- | --- | --- |
| Priority Queue Implementation Time Complexities | | |
| Function Name | **Time Complexity** | **Reason** |
| enqueue(newTask) | O(n) | While loop in the \_enqueue\_internal() function. |
| \_enqueue\_internal() | O(n) | While loop in the \_enqueue\_internal() function. |
| dequeue() | O(1) | This always removes the front. |
| peek() | O(1) | Checks what the first element is. |
| print() | O(n) | Traverses and prints all tasks. |
| is\_empty() | O(1) | Checks if the size = 0. |
| size() | O(1) | Returns the size of the queue. |
| contains(task) | O(n) | Does linear search to search for the task. |
| get\_priority(task) | O(n) | Does linear search to search for the priority of the given task. |
| remove\_task(task) | O(n) | Searching the task and removal takes O(n) complexity. |
| get\_task(task) | O(n) | Does linear search by the task name. |
| update\_priority(task, priority) | O(n) | Searching for the task and removal. |
| def \_get\_task\_internal(task) | O(n) | While loop. |
| get\_task\_index(task) | O(n) | While loop in the internal function. |
| \_get\_task\_index\_internal | O(n) | While loop. |
| clear(self) | O(1) | Uses the linked list class’s clear function. |
| \_get\_priority\_internal( task\_content) | O(n) | While loop. |
| \_contains\_internal(task) | O(n) | While loop. |