Data Structure:

Task Class: Create a Task class to represent individual tasks within a project. The Task class should contain properties such as task Name, description and status (e.g. in progress, completed).

Linked List Class: Implement a custom linked list class specifically designed to manage tasks. This linked list class should have methods for adding tasks, removing tasks, updating task statuses, and other operations necessary for efficient task management.

Algorithms:

Adding a Task: Implement a method in the linked list class to add a task to the appropriate position within the linked list based on the priority or order of tasks in the project.

Removing a Task: Create a method to remove a task from the linked list based on the task ID or other unique identifier.

Updating Task Status: Include a method to update the status of a task (e.g from in progress to completed) within the linked list.

Manipulating Tasks: Implement algorithms for manipulating tasks within the linked list, such as reordering tasks, moving tasks between projects, or other custom operations required by the project management company.

Advantages of Using Linked Lists:

Dynamic Size: Linked lists allow for dynamic resizing, making it easier to add or remove tasks without needing to pre-allocate memory for a fixed number of tasks.

Efficient Insertion and Deletion: The ability to efficiently add or remove tasks from any position within the list is crucial for managing tasks as projects evolve.

Sequential Order: Linked lists naturally lend themselves to managing tasks in sequential order, enabling team members to easily visualize and track project progress.

Flexibility: Linked lists provide flexibility in managing tasks, allowing for easy rearrangement and manipulation of tasks as project requirements change.