

Report of Backend API

Algorithms used:

1. SortTasks function:

- 1) **Sorting Algorithm:** This function uses the JavaScript Array `sort` method which is a variant of the quicksort algorithm.
 - When `sortBy` is 'asc', it sorts tasks by their `id` in ascending order.
 - When `sortBy` is 'desc', it sorts tasks by their `id` in descending order.

2. Authentication function:

- 1) **Token Verification Algorithm:** This function uses the `jwt.verify` method from the `jsonwebtoken` library to verify the JWT token. The algorithm checks the validity of the token against the secret key.
 - If the token is invalid or expired, an error response is sent.
 - If valid, the user information is attached to the request object, and the next middleware is called.

3. GET/tasks endpoint:

- **Filtering Algorithm:** Uses the `filter` method to filter tasks based on `filterBy` and `filterValue`.
- **Sorting Algorithm:** Calls the `sortTasks` function to sort the filtered tasks.
- **Pagination Algorithm:** Slices the filtered and sorted tasks array based on the `page` and `limit` query parameters using the `slice` method.

4. GET/tasks/:id endpoint:

- **Search Algorithm:** Uses the `find` method to locate the task with the specified `id` within the tasks array.

5. POST/tasks endpoint:

- **Basic Validation:** Checks if the `title` and `description` fields are present in the request body.
- **Insertion Algorithm:** Adds the new task to the `tasks` array using the `push` method.

6. PUT/tasks/:id endpoint:

- **Search Algorithm:** Uses the `find` method to locate the task with the specified `id`.
- **Update Algorithm:** Updates the title and description of the found task based on the request body.

7. DELETE/tasks/:id endpoint:

- **Search Algorithm:** Uses the `find` method to locate the task with the specified `id`.
- **Deletion Algorithm:** Uses the `splice` method to remove the found task from the `tasks` array.