```python
from google.colab import drive

# Mount Google Drive
drive.mount('/content/gdrive')

class TaskScheduler:
    def __init__(self):
        self.tasks = []
        self.filename = '/content/gdrive/My Drive/tasks.txt'

    def load_tasks(self):
        try:
            with open(self.filename, 'r') as file:
                lines = file.readlines()
                self.tasks = [line.strip().split(',') for line in lines]
        except FileNotFoundError:
            self.tasks = []

    def save_tasks(self):
        with open(self.filename, 'w') as file:
            for task in self.tasks:
                file.write(','.join(task) + '\n')

    def add_task(self, description, due_date, priority):
        new_task = [description, due_date, priority]
        if self.task_exists(new_task):
            print("A task with the same name and date already exists. Task not added.")
        else:
            self.tasks.append(new_task)
            self.save_tasks()
            print("Task added successfully!")

    def task_exists(self, new_task):
        for task in self.tasks:
            if task[0] == new_task[0] and task[1] == new_task[1]:
                return True
        return False

    def view_task_list(self):
        if not self.tasks:
            print("No tasks found.")
        else:
            for idx, task in enumerate(self.tasks, start=1):
                print(f"{idx}. {task[0]} - Due: {task[1]} - Priority: {task[2]}")

    def sort_tasks(self, sort_option):
        if sort_option == '1':
            self.tasks.sort(key=lambda x: x[2])
        elif sort_option == '2':
            self.tasks.sort(key=lambda x: x[1])
        else:
            print("Invalid sort option. Use '1' for priority or '2' for date.")
            return

        self.save_tasks()
        print(f"Tasks sorted successfully!")

    def display_sorted_tasks(self):
        if not self.tasks:
            print("No tasks found.")
        else:
            print("Sorted Task List:")
            for idx, task in enumerate(self.tasks, start=1):
                print(f"{idx}. {task[0]} - Due: {task[1]} - Priority: {task[2]}")

    def complete_task(self, task_index):
        if 1 <= task_index <= len(self.tasks):
            completed_task = self.tasks.pop(task_index - 1)
            self.save_tasks()
            print(f"Task '{completed_task[0]}' completed and removed from the list.")
        else:
            print("Invalid task index.")

    def edit_task(self, task_index, new_description, new_due_date, new_priority):
        if 1 <= task_index <= len(self.tasks):
            self.tasks[task_index - 1] = [new_description, new_due_date, new_priority]
            self.save_tasks()
            print("Task edited successfully!")
        else:
            print("Invalid task index.")

    def delete_task(self, task_index):
        if 1 <= task_index <= len(self.tasks):
            deleted_task = self.tasks.pop(task_index - 1)
            self.save_tasks()
            print(f"Task '{deleted_task[0]}' deleted successfully.")
        else:
            print("Invalid task index.")

    def export_tasks(self):
        with open(self.filename, 'w') as file:
            for task in self.tasks:
                file.write(','.join(task) + '\n')
        print(f"Tasks exported to '{self.filename}' successfully!")

    def search_task(self, keyword):
        matching_tasks = [task for task in self.tasks if keyword.lower() in task[0].lower()]
        if matching_tasks:
            for idx, task in enumerate(matching_tasks, start=1):
                print(f"{idx}. {task[0]} - Due: {task[1]} - Priority: {task[2]}")
        else:
            print(f"No tasks found matching the keyword '{keyword}'.")

# Initialize and run the task scheduler
task_scheduler = TaskScheduler()
task_scheduler.load_tasks()

while True:
    print("\nTask Scheduler\n--------------")
    print("1. Add a Task\n2. View Task List\n3. Sort Tasks\n4. Complete a Task\n5. Edit Task\n"
          "6. Delete Task\n7. Export Tasks\n8. Search Task\n9. Exit")

    choice = input("Enter your choice (1-9): ")

    if choice == '1':
        description = input("Enter task description: ")
        due_date = input("Enter due date (YYYY-MM-DD): ")
        priority = input("Enter priority (1-9): ")
        task_scheduler.add_task(description, due_date, priority)

    elif choice == '2':
        task_scheduler.view_task_list()

    elif choice == '3':
        sort_option = input("Choose sorting option:\n1. Sort by Priority\n2. Sort by Date\nEnter option (1 or 2): ")
        task_scheduler.sort_tasks(sort_option)
        task_scheduler.display_sorted_tasks()

    elif choice == '4':
        task_index = int(input("Enter the index of the task to complete: "))
        task_scheduler.complete_task(task_index)

    elif choice == '5':
        task_scheduler.view_task_list()
        task_index = int(input("Enter the index of the task to edit: "))
        new_description = input("Enter new task description: ")
        new_due_date = input("Enter new due date (YYYY-MM-DD): ")
        new_priority = input("Enter new priority (1-9): ")
        task_scheduler.edit_task(task_index, new_description, new_due_date, new_priority)

    elif choice == '6':
        task_scheduler.view_task_list()
        task_index = int(input("Enter the index of the task to delete: "))
        task_scheduler.delete_task(task_index)

    elif choice == '7':
        task_scheduler.export_tasks()

    elif choice == '8':
        keyword = input("Enter the keyword to search for in task descriptions: ")
        task_scheduler.search_task(keyword)

    elif choice == '9':
        print("Exiting Task Scheduler.")
```

```
            print("Exiting Task Scheduler.")
            break

        else:
            print("Invalid choice. Please enter a number between 1 and 9.")
```

```
Mounted at /content/gdrive

Task Scheduler
--------------
1. Add a Task
2. View Task List
3. Sort Tasks
4. Complete a Task
5. Edit Task
6. Delete Task
7. Export Tasks
8. Search Task
9. Exit
Enter your choice (1-9): 1
Enter task description: walking
Enter due date (YYYY-MM-DD): 2023-11-15
Enter priority (1-9): 3
Task added successfully!

Task Scheduler
--------------
1. Add a Task
2. View Task List
3. Sort Tasks
4. Complete a Task
5. Edit Task
6. Delete Task
7. Export Tasks
8. Search Task
9. Exit
Enter your choice (1-9): 2
1. home - Due: 23-11-10 - Priority: 5
2. project submit - Due: 23-11-12 - Priority: 1
3. reading - Due: 23-11-14 - Priority: 1
4. walking - Due: 2023-11-15 - Priority: 3

Task Scheduler
--------------
1. Add a Task
2. View Task List
3. Sort Tasks
4. Complete a Task
5. Edit Task
6. Delete Task
7. Export Tasks
8. Search Task
9. Exit
Enter your choice (1-9): 3
Choose sorting option:
1. Sort by Priority
2. Sort by Date
Enter option (1 or 2): 1
Tasks sorted successfully!
Sorted Task List:
1. project submit - Due: 23-11-12 - Priority: 1
2. reading - Due: 23-11-14 - Priority: 1
3. walking - Due: 2023-11-15 - Priority: 3
4. home - Due: 23-11-10 - Priority: 5
```