**GitHub Project Setup and Final Deliverables Roles**

**Team Name:** The Searchers

**Team Members:** Blake McGahee, Thiago Ramirez de Arellano, Gerson Diaz

**1. GitHub Project Setup**

The assignment explicitly directs us to use the **"Project 3 Template repository."** This is the best starting point as it likely contains a pre-configured CMakeLists.txt and a basic structure.

**Steps for GitHub Setup:**

1. **Create Your Repository from Template:**
   * Go to the provided "Project 3 Template repository" link.
   * Look for a button like "Use this template" (usually green, near the top-right).
   * Click it and choose "Create a new repository."
   * Name your repository something clear, like the-searchers-final-project or efficient-data-retrieval.
   * Ensure it's set to **Public** (or Private if you prefer, but remember to add the TA/Instructor as collaborators if private).
2. **Clone the Repository Locally:**
   * Once your new repository is created on GitHub, clone it to your local machine.
   * git clone [YOUR\_REPO\_URL]
   * cd [YOUR\_REPO\_NAME]
3. **Branching Strategy (Recommended for Teams):**
   * **main (or master) branch:** This branch should always contain stable, working code.
   * **Feature Branches:** Each team member should create a new branch for their specific tasks (e.g., blake/data-gen-jump-search, thiago/interpolation-search, gerson/ui-integration).
     + git checkout -b blake/data-gen-jump-search
   * **Work on Your Branch:** Make changes, commit frequently.
   * **Pull Requests (PRs):** When a feature is complete and tested on a feature branch, open a Pull Request to merge it into main. Team members should review each other's PRs before merging. This ensures code quality and collaboration.
4. **Initial Code Integration:**
   * Place your ProjectUtils.h (containing Blake's code and placeholders for Thiago's) into the src/ directory of the cloned template.
   * Gerson will create main.cpp in src/ and include ProjectUtils.h.
   * Ensure your CMakeLists.txt (from the template) is updated to include ProjectUtils.h and any corresponding .cpp files in the add\_executable commands for both Main and Tests targets, as discussed previously.
5. **Regular Commits and Pushes:**
   * Commit your changes regularly with descriptive messages (git commit -m "Implemented Jump Search algorithm").
   * Push your changes to your remote branch frequently (git push origin blake/data-gen-jump-search).
6. **README.md File:**
   * The template likely has one. Update it immediately with your team name, project title, and basic setup instructions. This will be an ongoing task throughout the project.

**2. Roles and Responsibilities for Final Deliverables**

Here's a breakdown of responsibilities for the final deliverables, ensuring all assignment criteria are met and leveraging your team's strengths:

**A. Group Deliverable (8% of grade)**

**1. Report (Maximum 5 pages PDF)**

* **Overall Responsibility:** Gerson (as the UI/Integration lead, will compile the final document). All members contribute their respective sections.
* **Administrative Section:**
  + **All Members:** Provide their GitHub usernames.
  + **Gerson:** Ensure Team Name, GitHub Repo Link, and Video Demo Link are correctly placed.
* **Extended and Refined Proposal (Suggested 2 Pages):**
  + **Problem, Motivation, Features, Description of Data, Tools/Languages/APIs/Libraries used:**
    - **All Members:** Review and refine the existing proposal content.
  + **Algorithms Implemented:**
    - **Blake:** Write detailed descriptions for **Jump Search** (its logic, how it works).
    - **Thiago:** Write detailed descriptions for **Interpolation Search** (its logic, how it works).
  + **Additional Data Structures/Algorithms used:**
    - **Blake:** Describe std::vector as the data storage and std::sort for initial sorting.
  + **Distribution of Responsibility and Roles: Who did what?**
    - **All Members:** Jointly update this section to reflect actual contributions during implementation.
* **Analysis (Suggested 1.5 Pages):**
  + **Any changes after proposal & rationale:**
    - **All Members:** Discuss any deviations from the proposal (e.g., specific implementation choices, minor feature changes) and document the reasons.
  + **Big O worst case time complexity analysis of major functions/features:**
    - **Blake:** Analyze generateAndSortDataset (dominated by std::sort's O(N log N)) and jumpSearch (O(N​) worst case).
    - **Thiago:** Analyze interpolationSearch (O(log log N) average, O(N) worst case).
    - **All Members:** Collaborate on analyzing the overall application's time complexity for core operations.
* **Reflection (Suggested 1-1.5 Page):**
  + **All Members:** Jointly discuss and write about the overall experience, challenges faced (e.g., debugging, Git conflicts), and what changes would be made to the project/workflow if starting over.
  + **All Members:** Individually contribute a paragraph or two on what *each* member learned through the process.
* **References:**
  + **All Members:** Contribute any specific academic papers, articles, or resources used for their respective algorithms/implementations.
  + **Gerson:** Compile the final list.
* **Formatting & Page Limit:**
  + **All Members:** Be mindful of the 5-page limit and 12pt font.
  + **Gerson:** Final check for adherence before PDF conversion.

**2. Source Code (2% of grade)**

* **Overall Responsibility:** All members are equally responsible for code quality, completion, and correctness.
* **Contributions:**
  + **Blake:** Ensure generateAndSortDataset and jumpSearch are robust, well-commented, and correct.
  + **Thiago:** Ensure interpolationSearch and the search query/result verification logic are robust, well-commented, and correct.
  + **Gerson:** Ensure the main UI loop, input/output formatting, and overall integration of the components are robust, well-commented, and correct.
* **GitHub Repository Link:**
  + **Gerson:** Ensure the correct public GitHub Repository URL is included in the Report.
* **README.md File:**
  + **Gerson:** Lead the creation and maintenance of the README.md with clear instructions for running the code.
  + **All Members:** Contribute details about their specific code units and any dependencies.

**3. Video (3% of grade)**

* **Overall Responsibility:** All members are responsible for the content and presentation quality.
* **Script/Content:**
  + **All Members:** Collaborate on writing a script that walks through all features, explaining each part.
  + **Blake:** Prepare to explain Data Generation & Sorting, and demonstrate Jump Search.
  + **Thiago:** Prepare to explain Interpolation Search and its demonstration.
  + **Gerson:** Prepare to explain the overall UI flow, how inputs are handled, and how results are displayed.
* **Recording & Editing:**
  + **Gerson:** Lead the recording (using Loom/Zoom) and basic editing to ensure it's within the 5-minute time limit and flows well.
* **Presentation:**
  + **All Members:** Decide if one person presents the whole project or if each person explains their part. Regardless, all should be familiar with the entire project.

**B. Individual Deliverables**

* **Peer Feedback (1% of grade):**
  + **All Members:** Each individual is responsible for providing constructive peer feedback for their two teammates.
* **Group Feedback (1% of grade):**
  + **All Members:** Collaborate as a group to complete and submit one group feedback document.

This detailed plan should help our team effectively manage the final project deliverables. Remember to communicate frequently and use GitHub's features (branches, pull requests) to keep our work organized!