-Contributors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Username | Email | Role | Contributions |
| Danny A. Hasen | Danny Hasen |  |  | * Contribution 1 * Contribution 2 |
| Jacob Carrasco | Gary Fox | garycarrasco@cpp.edu | Project manager | * Created repository * Added BagInterface * Started Efficiency Document * Added Intersection implementation * Implemented ArrayBag * Bug Fixes * Organized meetings * Created BagDriver |
| Joseluis Ramierz | Pizel29-wq |  |  | * Contribution 1 * Contribution 2 |

# -Repository Link

https://github.com/Gary-Fox/BagsProject

# -Table 1. The time complexities of this assignment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ResizableArrayBag | | | LinkedBag | | |
| Union | Intersection | Difference | Union | Intersection | Difference |
| Best case |  | One or both bags are empty, in which case an empty bag interface is returned. O(1) |  |  | One or both bags are empty, in which case an empty bag interface is returned. O(1) |  |
| Worst Case |  | We traverse only one of the bags (O(n) work), the frequency of the item is considered when updating both resultFrequency and the bags themselves (O(1\*resultFrequency))+(O1\*resultFrequency). The result frequency is always <= n, so absolute worst case we’re dealing with O(n + n + n), Therefore, our total time complexity is O(3n), or asymptotically **O(n)** |  |  | We traverse the contents of one bag (O(n) work) for every 1 item in the other bag. (O(n) work). In total, we do **O(n2)** work |  |