|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Homework 1 | - | 75/100 |  | [-5] You should shift and resize at the same time to be more efficient; [-2] Your exception messages should state exactly what is wrong, not just restate the name of the exception; [-4] removeFromBack and removeFromFront decrements size when the list is empty; [-5] clear does not reset the size of the array; [-9] When regrowing, you should regrow to twice the size of the array -Mayhul |  |
|  | Homework 10 | - | 88/100 |  | [-7] Prims fails for single vertex and large graph (should not check weight); [-5] BFS inefficient, should check that visited set is same size as # of vertices to terminate loop -Alex |
|  | Homework 2 | - | 78/100 |  | [-8] removeFromFront and removeFromBack fail, don't take into account the circular pointer or assign it wrong; [-2] removeLastOccurence fails because of remove function failures; [-2] Should say what the range is for indexoutofbounds; [-5] addToFront and addToBack should be O(1), use the trick taught in lecture/recitation and not loop through list; [-5] removeLastOccurence is inefficient, do not have to traverse the list twice, can keep track of the node and remove it in one traversal -Alex |
|  | Homework 3 | - | 100/100 |  | Perfect! s( ^ ‿ ^)-b - Mayhul |
|  | Homework 4 | - | 93/100 |  | [-1] distanceBetween fails on empty; [-1] remove decreases size before nosuchelem; [-5] DistanceBetween is inefficient, you have to loop through the path to the nodes twice to generate the distance -Alex |
|  | Homework 5 | - | 98/100 |  | [-2] Style (-2: Checkstyle) - Good job! ໒( ՞ ヮ ՞ )७ - Mayhul |
|  | Homework 6 | - | 92/100 |  | [-5] Calling put in resize is inefficient, it checks for duplicate keys when there are none; [-3] Generics problems -Alex |
|  | Homework 7 | - | 87/100 |  | [-5] Efficiency: Equals always traverses down both the left and right subtree, even if the left subtree returns false; [-3] Rotations should be done while removing the successor because the successor gets removed; [-2] Method not implemented recursively: secondLargest should be recursive; [-1] equals doesn't return the recursive call; [-1] removeH fails when node is null; [-1] Generics issues: You should cast to AVL<T> in equals - Mayhul |
|  | Homework 8 | - | 78/100 |  | [-5] mergeSort not stable, at 312 should be <= to take left element when equal; [-5] QuickSort too many comparisons, check 1 element case; [-1] Checkstyle; [-1] Javadocs not descriptive; [-5] Calling a pow function is inefficient, can keep just have a base variable that is multiplied by 10 every iteration; [-5] Cannot use Math.log10 -Alex |
|  | Homework 9 | - | 100/100 |  | Perfect! (ᵔᴥᵔ) - Mayhul |