A white paper with black text

Description automatically generated

A paper with text and images

Description automatically generated

Insertion Sort (Array)

public class InsertionSort {

    public static int[] insert(int[] a) {

        for (int j = 1; j < a.length; j++) {

            int key = a[j];

            int i = j - 1;

            while (i >= 0 && a[i] > key) {

                a[i + 1] = a[i];

                i = i - 1;

            }

            a[i + 1] = key;

        }

        return a;

    }

    public static void main(String[] args) {

        int[] a = {5, 2, 4, 6, 1, 3};

        int[] b = insert(a);

        for (int i = 0; i < b.length; i++) {

            System.out.println(b[i]);

        }

    }

}

Screenshot of code with correct output:

A screenshot of a computer program

Description automatically generated

Insertion Sort (Linked List) Optimization

// this optimization utilizes a LinkedList instead of an Array for InsertionSort

import java.util.LinkedList;

public class InsertionSortOptimization{

    public static void insertion\_sort(LinkedList<Integer> A){

        for(int j = 1; j < A.size(); j++){

            int key = A.get(j);

            int i = j-1;

            while(i >= 0 && A.get(i) > key){

                A.set(i+1, A.get(i));

                i = i-1;

            }

            A.set(i+1, key);

        }

    }

    public static void main(String[] args){

        LinkedList<Integer> A = new LinkedList<Integer>();

        A.add(5);

        A.add(2);

        A.add(4);

        A.add(6);

        A.add(1);

        A.add(3);

        insertion\_sort(A);

        for(int i = 0; i < A.size(); i++){

            System.out.println(A.get(i));

        }

    }

}

Screenshot of optimization working with correct output:

A screenshot of a computer program

Description automatically generated

**ALL CODE WORKS PROPERLY!!**