Example Class Hierarchy for a Sort Comparison System (Part 3)

AbstractSort

string CLASS\_NAME;

virtual void sort() = 0;

virtual void sort(array, size);

virtual void swap(val, val);

virtual int \* merge(a, b);

virtual void print() const;

virtual int getNumLoops() const {return 0;}

virtual int getNumInnerLoops() const {return 0;}

virtual int getNumOuterLoops() const { return 0;}

virtual int getNumSwaps() const {return 0;}

virtual int getNumMerges() const {return 0;}

virtual int getNumRecursions() const {return 0;}

virtual int getNumDivisions() const {return 0;}

string getSortType() const {return CLASS\_NAME;}

LinearSorts

int getNumLoops() const = 0;

int getNumOuterLoops() const {return 0;}

int getNumInnerLoops() const {return 0;}

int getNumSwaps() const {return 0;}

RadixSort

void sort();

void sort(int \* a, int arrSize);

void print() const;

int getNumLoops() const;

int getNumOuterLoops() const;

int getNumInnerLoops() const;

int getNumSwaps() const;

int \* getArr() const;

void init(array, size);

LinkedList

bool insert(int val);

// nothing else really needed for this application

Node \* nodes;

*composition*