PA04 - Sorting

Generated by Doxygen 1.8.6

Tue Nov 1 2016 13:24:03

Contents

1 Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

BubbleSort	**2**
MergeSort	**5**

2 File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

bubbleSort.cpp
Implementation file for BubbleSort class **10**

bubbleSort.h
Definition file for BubbleSort class **11**

mergeSort.cpp
Implementation file for MergeSort class **11**

mergeSort.h
Definition file for MergeSort class **12**

radixSort.cpp
Implementation file for RadixSort class **12**

radixSort.h
Definition file for radixSort class **13**

3 Class Documentation

3.1 BubbleSort Class Reference

Public Member Functions

- BubbleSort ()

  Constructor for class BubbleSort.
- ∼BubbleSort ()

  Destructor for class BubbleSort.
- void sort (int ∗data, int size)

  Sorts data from least to greatest.
- void swap (int ∗data, int &a, int &b)

  Function swaps two values.
- int **getComparisonNum** ()
- int **getSwapNum** ()

Private Attributes

- bool **sorted**
- int **numberOfComparisons**
- int **numberOfSwaps**

3.1.1 Constructor & Destructor Documentation

3.1.1.1 BubbleSort::BubbleSort ( )

Constructor for class BubbleSort.

Able to construct a BubbleSort object

Precondition

None

Postcondition

None

None

Parameters

None	

Exceptions

None	

Note

: None

3.1.1.2 BubbleSort::∼BubbleSort ( )

Destructor for class BubbleSort.

Able to destruct a BubbleSort object

Precondition

None

Postcondition

None

None

Parameters

None	

Exceptions

None	

Note

: None

3.1.2 Member Function Documentation

3.1.2.1 void BubbleSort::sort (int ∗ *data,* int *size*)

Sorts data from least to greatest.

Sorts data by comparing adjacent values over and over again and swapping them when out of order

Precondition

numberOfComparisons begins at 1
numberOfSwaps begins at 1

Postcondition

numberOfComparisons gets incremented every time function compares values
numberOfComparisons gets incremented every time function performs an action that moves the data around

Algorithm

Bubble Sort goes through a list of items over and over and compares the adjacent items. If the items are out of order, it swaps the two.

Parameters

data	This is the data the function will sort
size	The size of data

Exceptions

None	

Note

: None

3.1.2.2 void BubbleSort::swap (int ∗ *data,* int & *a,* int & *b*)

Function swaps two values.

Function swaps two values by storing one in a temp location

Precondition

int temp is intialized to b

Postcondition

temp gets stored in a

None

Parameters

in	*a*	gets b, which was stored in temp
in	*b*	gets a
in	*data*	contains a and b
out	*data*	a and b get swapped in this array

Exceptions

None	

Returns

None

Note

: None

The documentation for this class was generated from the following files:

- bubbleSort.h
- bubbleSort.cpp

3.2 MergeSort Class Reference

Public Member Functions

- MergeSort ()

  Constructor for class MergeSort.
- ∼MergeSort ()

  Destructor for class MergeSort.
- void sort (int ∗data, int first, int last)

  Sorts data from least to greatest.
- void merge (int ∗data, int first, int mid, int last)

  Merges data in the correct order.
- void **swap** (int ∗data, int &a, int &b)
- int **getComparisonNum** ()
- int **getSwapNum** ()
- void **reset** ()

Private Attributes

- int **numberOfComparisons**
- int **numberOfSwaps**

3.2.1 Constructor & Destructor Documentation

3.2.1.1 MergeSort::MergeSort ( )

Constructor for class MergeSort.

Able to construct a MergeSort object

Precondition

None

Postcondition

None

None

Parameters

None	

Exceptions

None	

Note

: None

3.2.1.2 MergeSort::∼MergeSort ( )

Destructor for class MergeSort.

Able to destruct a MergeSort object

Precondition

None

Postcondition

None

None

Parameters

None	

Exceptions

None	

Note

: None

3.2.2 Member Function Documentation

3.2.2.1 void MergeSort::merge (int ∗ *data,* int *first,* int *mid,* int *last*)

Merges data in the correct order.

Merges data from two separate arrays into a temp array and then copies the temp array into the final array

Precondition

None

Postcondition

None
numberOfComparisons gets incremented every time function performs an action that moves the data around

Algorithm

Merge simply takes data and splits it into two and then begins to merge it back together into a temp array. When it merges it compares the two arrays' first values and swaps them if necessary. It then places any remaining values in the temp array and then copies the temp array back into data with the resorted values.

Parameters

data	This is the data the function will merge
first	This is the index of the first value of the data
mid	This is the midpoint of the data
last	This is the index of the last value of the data

Exceptions

None	

Note

: None

3.2.2.2 void MergeSort::sort (int ∗ *data,* int *first,* int *last*)

Sorts data from least to greatest.

Sorts data by dividing the data over and over again until it is completely separated, and then merges the data in the correct order

Precondition

numberOfComparisons begins at 1
numberOfSwaps begins at 1

Postcondition

numberOfComparisons gets incremented every time function compares values
numberOfComparisons gets incremented every time function performs an action that moves the data around

Algorithm

Merge Sort is a recursive function that divides the data into two and then recursively calls itself with the divided data. It then merges the data back together in the correct order using the merge algorithm.

Parameters

data	This is the data the function will sort
first	This is the index of the first value of the data (begins at 0)
last	This is the index of the last value of the data (begins at the size • 1)

Exceptions

None	

Note

: None

The documentation for this class was generated from the following files:

- mergeSort.h
- mergeSort.cpp

3.3 RadixSort Class Reference

Public Member Functions

- RadixSort ()

  Constructor for class RadixSort.
- ∼RadixSort ()

  Destructor for class RadixSort.
- void sort (int ∗data, int size, int digits)

  Sorts data from least to greatest.
- int **getComparisonNum** ()
- int **getSwapNum** ()

Private Attributes

- int **numberOfComparisons**
- int **numberOfSwaps**

3.3.1 Constructor & Destructor Documentation

3.3.1.1 RadixSort::RadixSort ( )

Constructor for class RadixSort.

Able to construct a RadixSort object

Precondition

None

Postcondition

None

None

Parameters

None	

Exceptions

None	

Note

: None

3.3.1.2 RadixSort::∼RadixSort ( )

Destructor for class RadixSort.

Able to destruct a RadixSort object

Precondition

None

Postcondition

None

None

Parameters

None	

Exceptions

None	

Note

: None

3.3.2 Member Function Documentation

3.3.2.1 void RadixSort::sort (int ∗ *data,* int *size,* int *digits*)

Sorts data from least to greatest.

Sorts data by grouping it together by the data's speicific digits

Precondition

numberOfComparisons begins at 1
numberOfSwaps begins at 1

Postcondition

numberOfComparisons gets incremented every time function compares values
numberOfComparisons gets incremented every time function performs an action that moves the data around

Algorithm

Radix sort is a complex algorithm that groups the data together first by the least significant digit and keeps going all the way until it reaches the most significant digit. When it groups them together, it puts them in groups of 0s - 9s, all depending on what the digit is in that specific place. It then replaces them back into the array and moves on to the next least significant bit. By the time it is done with the most significant bit, all of the items are sorted.

Parameters

data	This is the data the function will sort
size	The size of data
digits	The max amount of digits there can be in a single item

Exceptions

None	

Note

: None

The documentation for this class was generated from the following files:

- radixSort.h
- radixSort.cpp

4 File Documentation

4.1 bubbleSort.cpp File Reference

Implementation file for BubbleSort class.

```
#include "bubbleSort.h"
```

4.1.1 Detailed Description

Implementation file for BubbleSort class.

Author

Alex Kastanek

Implements all member methods of the BubbleSort class

Version

1.00 C.S. Student (1 November 2016) Initial development and testing of BubbleSort class

Note

Requires bubbleSort.h
None

4.2 bubbleSort.h File Reference

Definition file for BubbleSort class.

```
#include <iostream>
```

Classes

- class BubbleSort

4.2.1 Detailed Description

Definition file for BubbleSort class.

Author

Alex Kastanek

Specifies all member methods of the BubbleSort class

Version

1.00 C.S. Student (1 November 2016) Initial development and testing of BubbleSort class

Note

None

4.3 mergeSort.cpp File Reference

Implementation file for MergeSort class.

```
#include "mergeSort.h"
```

4.3.1 Detailed Description

Implementation file for MergeSort class.

Author

Alex Kastanek

Implements all member methods of the MergeSort class

Version

1.00 C.S. Student (1 November 2016) Initial development and testing of MergeSort class

Note

Requires mergeSort.h
None

4.4 mergeSort.h File Reference

Definition file for MergeSort class.

```
#include <iostream>
```

Classes

- class MergeSort

4.4.1 Detailed Description

Definition file for MergeSort class.

Author

Alex Kastanek

Specifies all member methods of the MergeSort class

Version

1.00 C.S. Student (1 November 2016) Initial development and testing of MergeSort class

Note

None

4.5 radixSort.cpp File Reference

Implementation file for RadixSort class.

```
#include "radixSort.h"
#include <vector>
```

4.5.1 Detailed Description

Implementation file for RadixSort class.

Author

Alex Kastanek

Implements all member methods of the RadixSort class

Version

1.00 C.S. Student (1 November 2016) Initial development and testing of RadixSort class

Note

Requires radixSort.h
None

4.6 radixSort.h File Reference

Definition file for radixSort class.

```
#include <iostream>
```

Classes

- class RadixSort

4.6.1 Detailed Description

Definition file for radixSort class.

Author

Alex Kastanek

Specifies all member methods of the radixSort class

Version

1.00 C.S. Student (1 November 2016) Initial development and testing of radixSort class

Note

None

Index