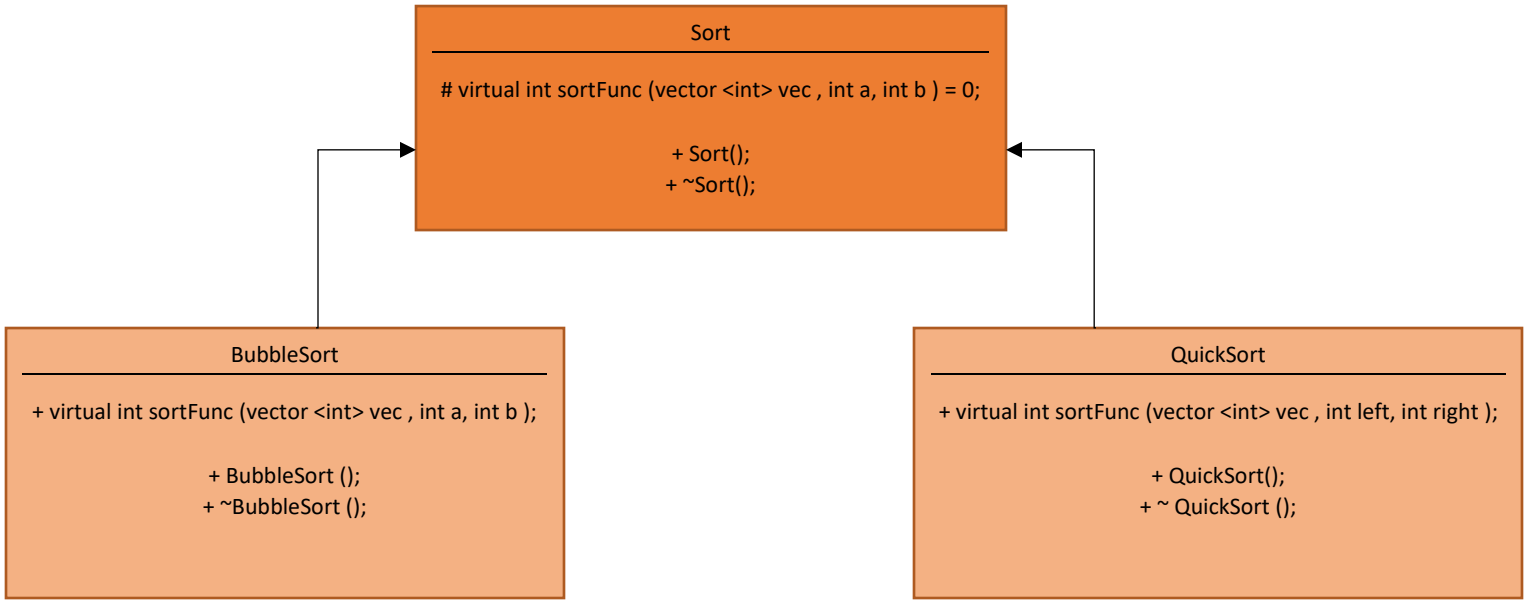
Renee Marsland
a1703500

Class Diagram

Sort



Sort

virtual int sortFunc
(vector <int> vec , int a, int b) = 0 : A pure virtual function that gets re-written for each of the different sorting functions derived from the base sorting class.

BubbleSort

int sortFunc
(vector <int> vec , int a, int b) : This function sorts the given vector using the 'Bubble Sort' method. Where it iterates through the array checking two integers at a time and switching their order according to which number is bigger.

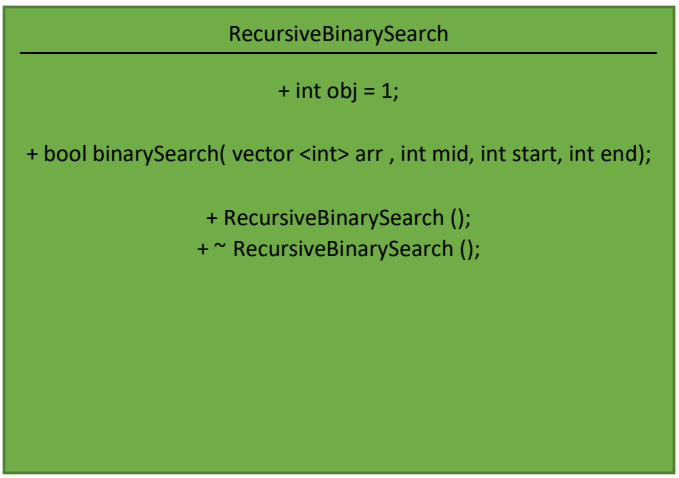
QuickSort

int sortFunc
(vector <int> vec , int left, int right) : This function sorts the given vector using the 'Quick Sort' method. It first divides the array into two smaller sub-arrays and then recursively sorts said sub arrays using the following steps:

1. Finds the middle of the array and calls that point the pivot.
2. Reorders the array so that all integers that are less than said pivot point get moved to the position before the pivot, and integers greater than the pivot get moved to the position before the pivot. If the integer is the same as the pivot it can go in either direction.
3. The above two steps are recursively applied to the sub-arrays until all integers are in their final position.

Renee Marsland
a1703500

RecursiveBinarySearch



int obj : For this program the value that is being searched for will always be '1', therefore I have chosen to hardcode it into the RecursiveBinarySearch function.

bool binarySearch
(vector <int> arr, int mid, int start, int end) :

Takes in the vector of integers being searched ('arr'), the midpoint of said vector, the starting location, and ending location of the vector.
The three integers are taken in so said elements can be altered for the recursion process of the function.
The way that this search function works is by looking at the midpoint of the vector and checking whether said midpoint equals the value that we are searching for.
If it does than the search is over and the function returns true. It also checks whether the end or start integers equals the value we are searching for and returns true if this is the case.
If the midpoint is greater than the searched for value, than we alter the midpoint and end point and re-search.
Similar occurs if the midpoint is less than the searched for values.

Renee Marsland
a1703500

Test Cases

Test Case ID	Input	Expected Output	Actual Output	Pass or Fail	Comments
01	“1 3 5 4 -5 100 7777 2014”	“true -5 1 3 4 5 100 2014 7777”	“ true -5 1 3 4 5 100 2014 7777”	P	To test whether it takes an input of values correctly (including spaces) and returns the expected output.
02	“0 3 5 4 -5 100 7777 2014“	“false -5 0 3 4 5 100 2014 7777”	“false -5 0 3 4 5 100 2014 7777”	P	To test whether it takes this particular set of numbers, including negatives and multiple digits, and successfully returns expected values.
03	“1 2 1 3 55 6 9 20”	“true 1 1 2 3 6 9 20 55”	“ true 1 1 2 3 6 9 20 55”	P	To test whether it takes an input of values correctly (including doubles of the same number) and returns the expected output.
04	“12 33 4 66 58 29 8”	“false 4 8 12 29 33 58 66”	“false 4 8 12 29 33 58 66”	P	Test whether it handles lots of double digit integers.