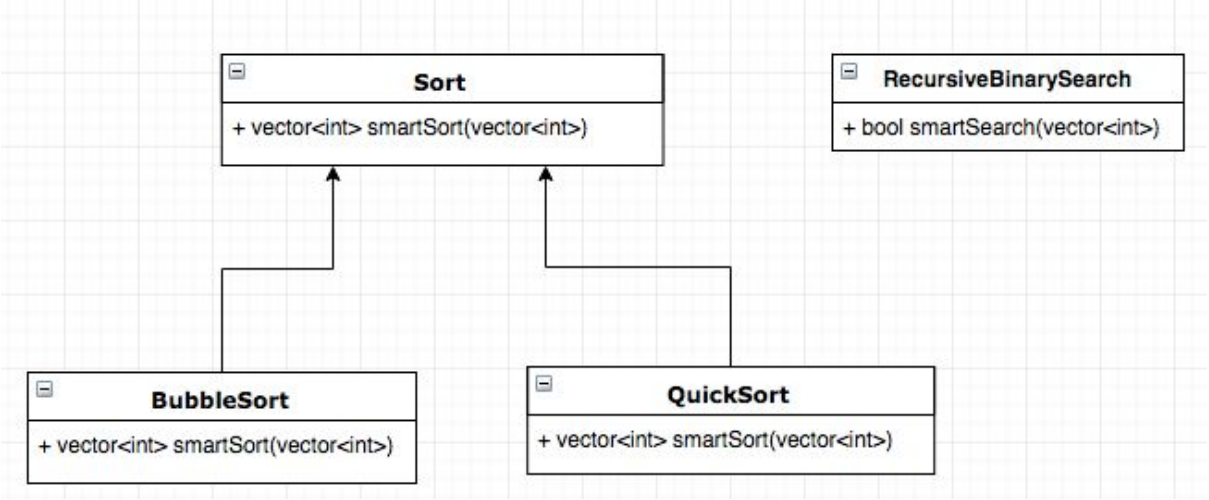
ADDS Prac 7 Design

Hideki Yoshinaga | a1658945

UML Diagram



Description

- Sort
  - smartSort : This function is a pure virtual function, takes a vector as its parameter, return a vector after sorting.

- BubbleSort
  - smartSort : This function is driven from the base class “sort”. takes a vector as its parameter, return a vector after sorting. The method of sorting is to compare two elements once in a list and swap the position if the first one is larger than the second. Once it reach to the end of the list, the last element can be determined to be the largest and we can then recursively do the same thing to the list-1 until list size is smaller than 2 then we can return the function.

- QuickSort
  - smartSort : This function is driven from the base class “sort”. takes a vector as its parameter, return a vector after sorting. The method of sorting is : first divides a large array into two smaller sub-arrays (elements less than the second element and elements greater or equal to the second element). Recursively apply the above steps to the sub-array until the sub-array is size 2. The base case of the recursion is arrays of size zero or one, which never need to be sorted.

- RecursiveBinarySearch

    - smartSearch : This function recursively find out if a given number is in the sorted array or not and return bool value. When the array is larger than 1, find the middle point of the array by dividing the array size by 2 and check the element in the middle is the given number or weather larger than the given number or not. If yes then return true, if larger then return its own function recursively with first half of the array given as parameter, same to if it is less than given number. else will return false.

- Main
    - Read one line input store it as a string, then separate out the first element to a string as it will decide which sorting method to use. the rest of the string are converted to int and stored in a vector. According to the first element, decide to create QuickSort object or BubbleSort object. Apply smartSort to the vector, use RecursiveBinarySearch to check if the vector has the given number and cout the result. After it, use a for loop to print all element in the vector.

Testing

- test integers in dessending order with BubbleSort
    - B 9 8 7 6 5 4 3 2 1 0 -1
    - true -1 0 1 2 3 4 5 6 7 8 9
- test integers in dessending order with QuickSort
    - Q 9 8 7 6 5 4 3 2 1 0 -1
    - true -1 0 1 2 3 4 5 6 7 8 9
- test with wrong input (first element is not B or Q)
    - 1 2 3 45 6 7 8 9 0
    - program break
- test with wrong input (element after the first one are not int)
    - B A D E F
    - program break
- test if RecursiveBinarySearch can reach first and last element
    - B 6 7 8 9 5 4 3 2 1
    - true 1 2 3 4 5 6 7 8 9
- test if RecursiveBinarySearch can reach first and last element
    - B 7 8 9 5 4 3 2 1 6
    - true 1 2 3 4 5 6 7 8 9
- test if the program can ignore more than one spaces
    - Q 3 4 5 8
    - false 3 4 5 8
- test if the program can arrange elements with same number
    - Q 1 1 1 1 1 1 1 1 1 1 1 1
    - false 1 1 1 1 1 1 1 1 1 1 1 1