Jake Goldstein

HW 05 Written

1. pseudo code for perform\_if
   1. Assign an itr to itrStart to use for a loop.
   2. Create an int that will count how many times the functors will run
   3. Loop through the data set
   4. Test each element with one of the functors
   5. If true run the other functor, and add one to the count
   6. Return the total count

Preconditions are that the iterator being passed has the ++operator overloaded, and the \*operator.

Postcondtions is an int that tells how many elements in the data set returned true for a given functor that was passed in.

Run time is O(n)

1. (The following tree did not come out as nicely as I had wanted it to but I think it gets the message across)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | 4 |  |  |  |
|  |  | 2 |  |  | 2 |  |  |
| 1 |  |  | 1 |  | 1 | 1 |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

b.

4: 2: 1: 0: 0:

1: 0: 0:

\*

2: 1: 0: 0:

1: 0: 0:

\*

\*\*\*

c. O(nlog(n))

1. a. 4

2

1

0

b. 4: 2: 1: 0:

c. O(log(n))

1. For n = 3 there are 5 function calls, and when n = 4 there are 9 function calls.
   1. I needed to see more of a pattern to get when n = 5. When n = 1, there is only one function call. n = 2 there are 3 function calls, and when n = 3 there are 5, n = 4 there are 9. So: 1, 3, 5, 9. These are (2\*fib(n) -1), so when n = 5 there will be 15 function calls.

8, 9, -11, 2, 0, 3

-11, 8, 9, 2, 0, 3

-11, 2, 8, 9, 0, 3

-11, 0, 2, 8, 9, 3

-11, 0, 2, 3, 8, 9

8, 9, -11, 2, 0, 3

-11, 8, 9, 2, 0, 3

-11, 8, 9, 0, 2, 3

-11, 8, 9, 0, 2, 3

-11, 0, 2, 3, 8, 9

-11, 8, 3, 2, 0, 9

-11, 0, 2, 9, 8, 3

-11, 0, 2, 3, 8, 9

1. Merge Sort:

28, 10, 2, 27, 5, 1

28, 10, 2 27, 5, 1

28, 10 2 27, 5 1

28 10 27 5

Quick Sort:

28, 10, 2, 27, 5, 1

1 28, 27, 5, 10

5, 10 28

10

1. Insertion : O(n)

Merge: O(nlog(n))

Quicksort: O(nlog(n))

1. O(nlog(n))
2. Because at that point you have already reached the end of the sorting.