Huimin Wang

Dr. Smallberg

CS 31 Lec 2

5 November 2017

Report

Notable Obstacles

I struggled with the last function split. In the beginning, I was only able to move all the strings less than the splitter to the front. I did this by swapping the positions of the smaller element with the first element in the array. After each swapping, I increment the position of the initial element. When the next smaller element is found, it is going to swap with the second element in the array. I thought I could use the same algorithm for the strings larger than the splitter, but it turned out I need to add more one layer of loop. With this layer of loop, the program always repeats sorting the strings from the beginning, which is the position of the first element that is not smaller than the splitter. Every time a bigger element is found, the rest of the elements move left one place, and the found element is moved to the end.

Test Data

1. An array with string h[7] = { "selina", "reed", "diana", "tony", "", "logan", "peter" };
2. An array with all the elements found in h: string h2[7] = { "selina", "reed", "diana", "tony", "", "logan", "peter" };
   * The subsequence function in my program initially did not handle this array correctly because the loop only iterates position 0 to n1-n2 in a1. It should iterate from 0 to n1.
3. An array with the first two elements equal to h: string g[4] = { "selina", "reed", "peter", "sue" };
4. An subsequence of h: string e[4] = { "diana", "tony", "", "logan" };
5. An array with repeated elements: string d[5] = { "reed", "reed", "reed", "tony", "tony" };
6. An array with no repeated elements: string f[3] = { "peter", "diana", "steve" };
7. An empty array: string k[0] = {};
   * The subsequence function in my program initially did not handle this array correctly because it is an empty array. I handle the situation by catching n = 0 in my function. When the function is passed an empty array, it will always return 0 because an empty array is a subsequence of every array.
8. An array with one single element (used to test subsequence): string l[1] = { "peter" };
9. An array of letters which is easy to see if the string is split correctly:

string x[9] = { "e", "h", "b", "d", "z", "x", "a", "a", "c" };

* The split function in my program initially did not handle this array correctly. The elements larger than the splitter are not at the end.

1. An array with 5 sequences:

string y[9] = { "tony", "bruce", "steve", "steve", "diana", "diana", "diana", "steve", "steve" };