

Kyle Ryan
CMPT435
24 March 2021

Assignment 5

1. Analyze the following code and provide a "Big-O" estimate of its running time in terms of n . Explain your analysis.

```
function fun(A[], n)
{
    if (n>1)
        fun(A[], n-1)
        j = n-1
        while (j>0)
            swap(A[j], A[j-1])
            j = j-1
        end while
    end if
}
```

Note: Credit will not be given only for answers - show all your work:

(4 points) steps you took to get your answer.

$$\begin{aligned} T(n) &= T(n-1) + n \\ &= T(n-2) + n-1 + n \\ &= T(n-3) + n-2 + n-1 + n \\ &\dots \\ &= n(n+1)/2 \\ &= O(n^2) \end{aligned}$$

I came to the above recurrence relation because the recursive call is $n-1$ and the while loop executes n times.

(1 points) your answer.

Resultantly, the runtime is $O(n^2)$

3. Input: A list of alphabetic characters.

Output: Write a recursive algorithm that determines whether or not the string “yes” occurs in X, and if so, to return the location of the rightmost occurrence of “yes”.

For example, given {'a', 'y', 'e', 's', 'w', 'y', 'e', 's', 'v', 'h', 'e'}, output: 5. Given {'o', 't', 'z', 'y', 'e', 's', 'v', 'g', 'r', 'a', 'y', 'e', 's'}, output: 10

Design a recursive algorithm to solve this problem (7 points).

(i) describe the idea behind your algorithm in English (1 points);

My algorithm will check to see if the current element in array A is equal to the current element in the strToChar array. If the element is not found then it will recursively call the function with j-=1 looking for where there will be the last occurrence of all the elements in the strToChar array.

(ii) provide pseudocode (4 points);

```
if (j >= A.length)
    return -1;

while (i > -1 && j > 0) {
    if (strToChar[i] == A[j]) {
        i--;
    }
    else
        where(A, s, j-=1);
    }
    return j;
}
```

(iii) analyze its running time (2 points).

$T(n) = T(n-1)$
 $= T(n)$
 $= O(n)$

Full credit (7 points) will be awarded for a recursive algorithm. Algorithms that are NOT recursive will be scored out of 2 points.