Homework 2 – S20201116 이수빈

**Finding the length of the Longest Common Sequence (LCS) problem**

Let’s define LCS of x[1,2,…i] and y[1,2,...j] as c[i,j]. Then,

c[i-1,j-1]+1 (if x[i]=y[i])

*<equation 1>*

c[I,j]

max{ c[i-1,j], c[i,j-1]} (else)

Our goal would be finding c[m,n] when length of array x is *m* and length of array y is *n*. Each c[i-1,j] and c[i,j-1] is subproblem of c[i,j] and is an optimal solution for its problem. Therefore, we can conclude that c[m,n] consists of overlapping subproblems which means its solution is consists of many solution of subproblems. Therefore, we can solve LCS as *recursive dynamic programming*.

To implement recursive dynamic programming, I first used the method of calling function itself recursively. However, it was too slow and produced stack overflow. Recursively calling function requires stack memory to save its address and value. However, the size of stack memory is limited and for problems like this (which input is 70k long each) would produce overhead. Because of overflow of stack memory, it produced segmentation fault error.

To solve this, I used *for loop* for this problem. Unlike the previous method, it does not return value but save it and add up somewhere. In my code, I used array *c* to keep track of values. *c* is first initialized to 0 at the beginning of each loop. Then, it follows the same step as *<equation 1>* as shown above.

**pseudo code**

MAX=max(arr1, arr2)

MIN=min(arr1,arr2)

c[MIN];

for i←0 to MAX

curr=0;

for j ← 0 to MIN

tmp=c[j]

if (i or j =0) c[j]=0;

else if (arr1[i-1]=arr2[j-1]) c[j]=curr+1;

else c[j]=max(c[j-1], c[j]);

curr=tmp;

return c last elemet;

**Time Complexity**: O(mn)

As shown in pseudo code above, code goes through 2 for loop which size is m and n. Therefore, its time complexity is m\*n.

**Space Complexity**: O(min{m,n})

Maximum length of LCS is min{m,n} since it cannot be larger than the length of array itself. So, array *c* is allocated with size of min{m,n} which is space complexity.

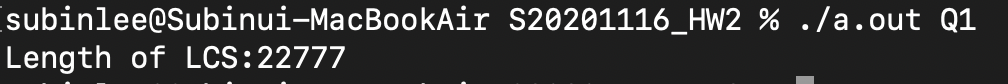
**Input**

two line consist of 70K letters

**Output**

*\* compiled execution file name: a.out*

*\* argument to pass : Q1 (file name)*



→22777