Time Spent: 2 hours

Collaborators and Resources: Talked with Nathaniel MacArthur-Warner about methodology and interpreting results

Problem 1

For this algorithm, which must run in O(n^c) time, we saw the arbitrary nature of the interleaving as a crucial factor in the design of the algorithm. Given an n-symbol string x and two other strings y and z, the goal is to describe whether x is a hybrid of y and z. Because of the uncertainty about whether the jth letter in x comes (in order) from the nth letter in y or the mth letter in z (or both), iterating through x once and just looking at the corresponding elements of its components y and z isn't necessarily going to work. In fact, our algorithm invokes a matrix which can just verify whether x is a hybrid based on the last element (in the bottom row and last column) . . .

```
1. Given an $n$-symbol string $x$ and two other strings $y$ and $z$
2. Initialize a matrix M with $|y|$ columns and $|z|$ rows such that $M_{0,0}$
   = $True$ and all other elements in M = $False$
3. For each unchecked $False$ element (i,j) of M
   (a) If $M_{i-1,j} == True$ or $M_{i,j-1} == True$
       # That is, if there's a valid path from the element above (i,j)
       or the element to the left of (i,j) to (i,j). (i,j) must border a
       $True$ value.
        i. If $x$[i+j] == $y$[i] then $M_{i,j} = True$
       ii. Else if $x$[i+j] == $z$[j] then $M_{i,j} = True$
           # If the i+jth element of $x$ is equal to the ith element of $y$
           (columns) or the jth element of $z$ (rows) then we continue
           the path
   (b) Else # If there are no longer any valid paths
        i. Break
```

4. If $M_{|y|,|z|} == True$
    (a) Return "x is a hybrid of y and z."
5. Else if $M_{|y|,|z|} == False$
    (a) Return "x is not a hybrid of y and z."

	-	W	O	R	L	D
-	$True_{\rightarrow}$	$True_{\rightarrow}$	$True_{\rightarrow}$	$True_{\downarrow}$	$False$	$False$
H	$False$	$False$	$False$	$True_{\rightarrow}$	$True_{\downarrow}$	$False$
E	$False$	$False$	$False$	$False$	$True_{\rightarrow}$	$True_{\downarrow}$
L	$False$	$False$	$False$	$True$	$False$	$True_{\downarrow}$
L	$False$	$False$	$False$	$False$	$False$	$True_{\downarrow}$
O	$False$	$True$	$False$	$False$	$False$	***True***

In the example given above we know that y[1:5] is WORLD and z[1:5] is HELLO. We can infer just from the final element in the matrix that the 10-symbol string "WORHLEDLLO" is composed of interleaving the characters of the two composite words.

Claim 1. The algorithm correctly returns whether x is a hybrid of y and z and does so in O(n^c) time for some constant c.

Proof. Certainly the algorithm works for finding whether the first letter of x ("W" in this case) can be found from the first letters of either y ("WORLD") or z ("HELLO").
It does so by checking whether $x[1] == y[1]$ (if so, $M_{1,0}$ becomes $True$) or whether $x[1] == z[1]$ (if so, $M_{0,1}$ becomes $True$). If neither of these are the case then the algorithm will simply break so we won't run into the next few cases. So it works for the first character, and we could hypothesize that it's going to work for the first n characters. Now, for the $n + 1^{st}$ character of x: The algorithm compares the $n + 1^{st}$ character of x to that of y and z as follows.

If x[n+1] == y[(n-j)+1] (and either the element above or to the left is $True$), then we have a valid path to x[n+1].

If x[n+1] == z[(n-i)+1] (and either the element above or to the left is $True$), then we have a valid path to x[n+1].

If x[n+1] == y[(n-j)+1] and x[n+1] == z[(n-i)+1] (and both elements above and left are $True$), then the algorithm still correctly marks the (k,j) entry (where k+j = n+1) as $True$. (That is, if the $n+1^{st}$ letter of x can be found as the $i+1^{st}$ letter of y and alternatively (possibly) the $j+1^{st}$ letter of z).

So by the principle of mathematical induction we know that the algorithm is going to successfully identify the correct path. It is able to do so for all characters in x if and only if $M_{len(y),len(z)} == True$. So by this metric we will be able to identify whether or not x is a hybrid or not.

Furthermore, the algorithm is going to run in O(4·$|y|$·$|z|$) $\leq$ O(4·($|y|$+$|z|$)^2) = O(($|y|$+$|z|$)^2) = O(n^2) where x has n symbols. This is because as nonzero integers if $|y|$ and $|z|$ are both less than n, their product is going to be less than n^2. Since we are making 4 comparisons for the total number of cells $|y| \cdot |z| \leq n^2$, we have to multiply by 4. However this is a constant coefficient which means that the runtime is O(n^c) where c = 2.

□