Edit Distances: Verification

Yi Wang

Department of Computer Science and Engineering
Chinese University of Hong Kong

October 23, 2019

Given two strings s, t, we already know how to compute their edit distance $edit(s, t)$ using dynamic programming in $O(|s||t|)$ time. It turns out that we can do better if we only need to verify whether $edit(s, t) \leq d$. This can be done in

$$O(|s| + |t| + d \cdot \min\{|s|, |t|\})$$

time.

We will consider only $|s| = |t| = \ell$. The case of $|s| \neq |t|$ is similar and left to you.

Our goal now is to verify whether $edit(s, t) \leq d$ in $O(d\ell)$ time for $d < \ell$ (if $d \geq \ell$, the answer is trivially yes).

Recall that, in order to compute $edit(s, t)$ in $O(\ell^2)$ time, our strategy was to fill in an $(\ell + 1) \times (\ell + 1)$ array A. To solve the verification problem, we will adopt a similar strategy, except that we will fill in only a hexagon part of A, as explained next.

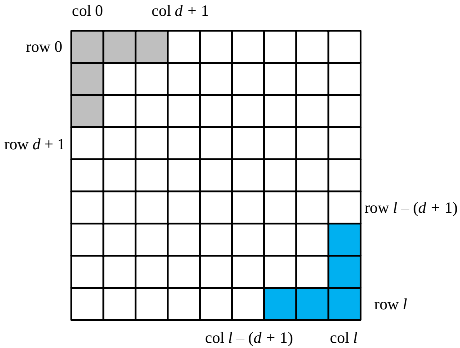
Let us first define the gray boundary cells to be

- At row 0, the left most $d + 1$ cells.
- At column 0, the top most $d + 1$ cells.

Define the blue boundary cells to be

- At row ℓ, the right most $d + 1$ cells.
- At column ℓ, the bottom most $d + 1$ cells.

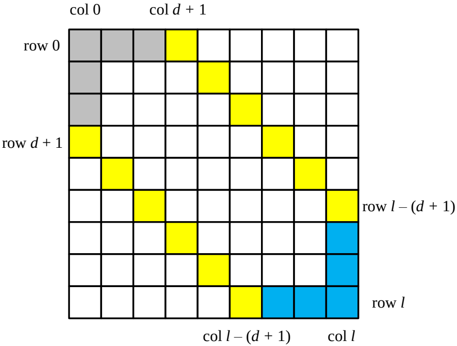
An example with $\ell = 8$ and $d = 2$:



Define the yellow boundary cells to be:

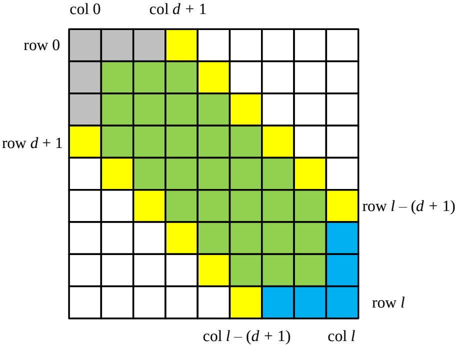
- $A[0, d+1]$, $A[1, d+2]$, ..., $A[\ell-(d+1), \ell]$
- $A[d+1, 0]$, $A[d+2, 1]$, ..., $A[\ell, \ell-(d+1)]$

An example with $\ell = 8$ and $d = 2$:



Define the green cells to be all those cells inside the region surrounded by the gray yellow, and blue boundary cells.

An example with $\ell = 8$ and $d = 2$:



We fill in only the colored cells (i.e., ignoring the others) as follows:

1. Fill in the gray cells normally.
2. Put $\geq d + 1$ in all the yellow cells.
3. Compute the green and blue cells in the same manner as in the $O(\ell^2)$-time algorithm (i.e., row by row, and left to right at each row).

Report yes if $A[\ell, \ell] \leq d$, and no, otherwise.

Since there are only $O(d\ell)$ colored cells, the running time is $O(d\ell)$.

Example: $s =$ `humanity`, $t =$ `hunamity`, and $d = 2$.

After the first two steps:

		h	u	m	a	n	i	t	y
	0	1	2	3					
h	1				$\geqslant 3$				
u	2					$\geqslant 3$			
n	3						$\geqslant 3$		
a		$\geqslant 3$						$\geqslant 3$	
m			$\geqslant 3$						$\geqslant 3$
i				$\geqslant 3$					
t					$\geqslant 3$				
y						$\geqslant 3$			

Edit distance by recurrence.

- If $m > 0$, $n > 0$, and $s[m] = t[n]$, then $edit(s, t)$ is:

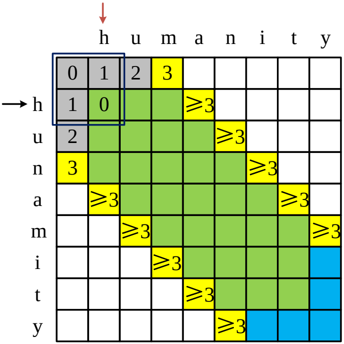
$$\min \begin{cases} 1 + edit(s, t[1..n-1]) \\ 1 + edit(s[1..m-1], t) \\ edit(s[1..m-1], t[1..n-1]) \end{cases} \tag{1}$$

- If $m > 0$, $n > 0$, and $s[m] \neq t[n]$, then $edit(s, t)$ is:

$$\min \begin{cases} 1 + edit(s, t[1..n-1]) \\ 1 + edit(s[1..m-1], t) \\ 1 + edit(s[1..m-1], t[1..n-1]) \end{cases} \tag{2}$$

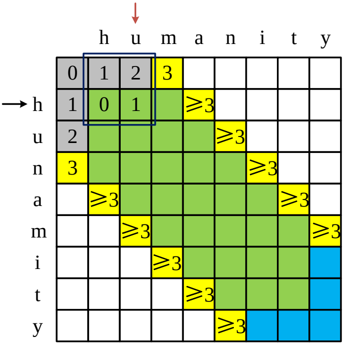
Example: $s = \texttt{humanity}$, $t = \texttt{hunamity}$, and $d = 2$.

One more step:



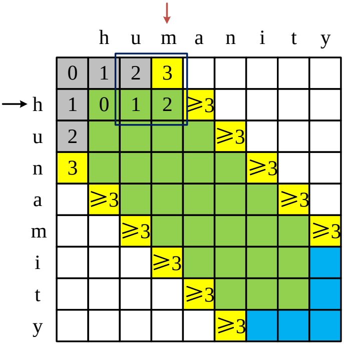
Example: $s = \texttt{humanity}$, $t = \texttt{hunamity}$, and $d = 2$.

One more step:



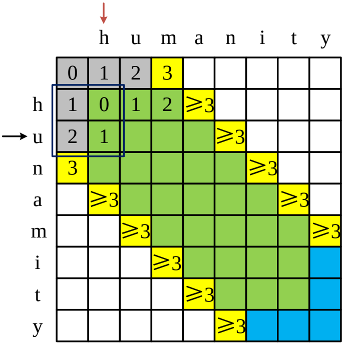
Example: $s = \texttt{humanity}$, $t = \texttt{hunamity}$, and $d = 2$.

One more step:



Example: $s = \texttt{humanity}$, $t = \texttt{hunamity}$, and $d = 2$.

One more step:



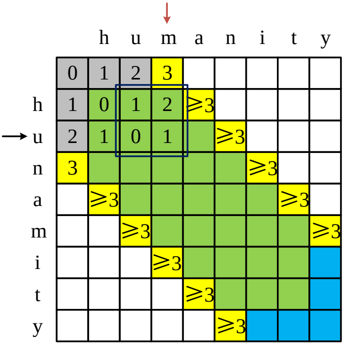
Example: $s = \texttt{humanity}$, $t = \texttt{hunamity}$, and $d = 2$.

One more step:

		h	u	m	a	n	i	t	y
	0	1	2	3					
h	1	0	1	2	$\geqslant 3$				
u	2	1	0			$\geqslant 3$			
n	3						$\geqslant 3$		
a		$\geqslant 3$						$\geqslant 3$	
m			$\geqslant 3$						$\geqslant 3$
i				$\geqslant 3$					
t					$\geqslant 3$				
y						$\geqslant 3$			

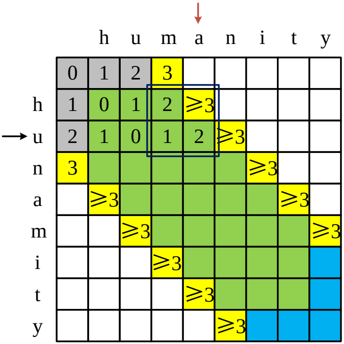
Example: $s = \texttt{humanity}$, $t = \texttt{hunamity}$, and $d = 2$.

One more step:



Example: $s =$ `humanity`, $t =$ `hunamity`, and $d = 2$.

One more step:



Example: $s = \texttt{humanity}$, $t = \texttt{hunamity}$, and $d = 2$.

After all steps:

		h	u	m	a	n	i	t	y
	0	1	2	3					
h	1	0	1	2	$\geqslant 3$				
u	2	1	0	1	2	$\geqslant 3$			
n	3	2	1	1	2	2	$\geqslant 3$		
a		$\geqslant 3$	2	2	1	2	3	$\geqslant 3$	
m			$\geqslant 3$	2	2	2	3	4	$\geqslant 3$
i				$\geqslant 3$	3	3	2	3	4
t					$\geqslant 3$	4	3	2	3
y						$\geqslant 3$	4	3	2

So we conclude $edit(s, t) \leq 2$.

Think

Why is the algorithm correct?