ADA Mini HW3

Written by CSIE b08902075 林耘平

First, let's solve this problem by using a 2-D table. Define $dp[i][j]$ as the cost of transforming the first i characters of S_a into the first j characters of S_b.

Trivially, the transition function for edge cases is

$$dp[i][j] = \begin{cases} 0 & i = 0,\ j = 0 \\ i \cdot e & i \neq 0,\ j = 0 \\ j \cdot d & i = 0,\ j \neq 0 \end{cases}$$

Consider the general case where $i \neq 0, j \neq 0$.

If $S_a[i] = S_b[j]$, then $dp[i][j]$ is at most $dp[i-1][j-1]$. If $S_a[i] \neq S_b$, then $dp[i][j]$ is at most $dp[i-1][j-1] + f$.

Combining with the cases that involve adding or removing, we can write down the transition function as

$$dp[i][j] = \min(dp[i-1][j] + e,\ dp[i][j-1] + d,\ dp[i-1][j-1] + (S_a[i] \neq S_b[j]) \cdot f).$$

Now we can try to reduce the space complexity from $O(ab)$ to $O(a)$.

Since in every transition, we only take the minimum from some elements in the current column and the last column. We can trim the table of size $a \cdot b$ into a table of size $a \cdot 2$ and reuse it in m iterations. Whenever we need the data from the last column, we can just read the data from the other column.

Now the definition of $dp[i][j\%2]$ in the j iteration is the cost of transforming the first i characters of S_a into the first j characters of S_b.

For $j = 0$, the table is set as

$$dp[i][0] = \begin{cases} 0 & i = 0 \\ i \cdot e & i \neq 0 \end{cases}$$

In the j iteration where $j > 0$, the transition function is

$$dp[i][j\%2] = \begin{cases} j \cdot d & i = 0 \\ min(dp[i-1][j\%2] + e,\ dp[i][(j+1)\%2] + d, dp[i-1][(j+1)\%2] + (S_a[i] \neq S_b[j]) \cdot f) & i \neq 0 \end{cases}$$

Since the time used for each (i, j) is $O(1)$ and space used is of size $a * 2$, we can transform S_a to S_b with $O(ab)$ time complexity and $O(a)$ space complexity. The answer is stored in $dp[a][b\%2]$.

However, we still need to transform S_b to S_a. Now define $dp[i][j]$ as the cost of transforming the first j characters of S_b to the first i characters of S_a.

Similarly, we have

$$dp[i][j] = \min(dp[i-1][j] + d,\ dp[i][j-1] + e,\ dp[i-1][j-1] + (S_a[i] \neq S_b[j]) \cdot f).$$

To reduce the space complexity to $O(a)$, let's reuse two columns just as we did before. Using a table of size $a \times 2$ will be enough. The answer is stored in $dp[a][b\%2]$ as well.

Transform S_a to S_b and the other way around, and take the minimum of the costs will yield the final answer to this problem.