

ALGORITHMS, FALL 2018, HOMEWORK 9

1. If we use a table to store whether or not there's a "path" to the letter in C .

We draw a table with the characters of S_1 along the top and the characters of S_2 along the side. We keep track of the letters we're currently at and fill the table in from top to bottom. Every square is required to have at least one square to the top or one square to the side as true, symbolizing that there exists a path to it through the blend.

This is $O(nk)$ complexity because the table is $O(nk)$ size.

2. If you use an array for this question, we can say that insertions are relatively cheap because we just stick it on the end of the array.

We can now exaggerate the cost of insertion to 2 per insertion, so that when we need to resize, we amortize the cost down to constant time again. To delete half, that will take $2n$ time, where n is the number of insertions, but because each insertion is exaggerated to cost 2 , we have $2n$ cost - $2n$ deletion cost, which becomes constant time again.