

1. Nucleotides

Нам необходимо было написать программу, принимающую на вход две последовательности символов и выводящую на экран последовательность минимальной длины, символы которой содержат в качестве подпоследовательностей две исходные.

Входные данные: На вход даются две строки A и B - это и есть исходные последовательности символов.

Выходные данные: Строка минимально возможной длины, содержащая в качестве подпоследовательностей символы строк A и B.

Алгоритм: Посчитаем расстояние Левенштейна между строками A и B через динамику, и запомним матрицу, которой мы пользовались при этом - создание и заполнение матрицы осуществляют функции `matrix`, `relax` и `mimm`. Теперь будем "идти" в обратную сторону. То есть изначально мы находимся в нижнем правом углу матрицы и движемся к верхнему правому, постепенно заполняя строку "gt иско- мую последовательность символов. Движение обеспечивает рекурсивная функция `forppu`.

Тесты:

GCCATACGGAGACTTTTCG + CCAATCTTCC = GCCAATACGGAGACTTCTCG

GCTATGAGCTGGCC + TGGCCCAACGTCGCATGTG = TGGCCTCAATCGTACGCATGTGCG

Вывод:

На самом деле, данная программа работает корректно только при наличии необходимого объема памяти для того, чтобы хранить матрицу размером $n*m$, где n и m - длины строк A и B соответственно. Время ее работы составляет $O(m*n)$ и не представляется возможным ее ускорение. Таким образом, основным минусом представленной программы является память, требуемая для хранения используемой матрицы. В итоге, имея необходимый объем памяти и достаточное количество времени, программа работает для строк любой конечной длины.