Динамика на последовательностях.

В рамках темы "динамика на последовательностях"обычно рассматривают 2 стандартные задачи:

- Поиск наибольшей общей подпоследовательности
- Поиск наибольшей возрастающей подпоследовательности

Формально условие задачи поиска наибольшей общей подпоследовательности можно сформулировать так: Даны 2 массива $a_1, a_2, ...a_n, b_1, b_2, ...b_m$ необходимо найти такие максимальные множества индексов $i_1 < i_2 < ... < i_k < n+1$ и $j_1 < j_2 < ... < j_k < m+1$, что $a_{i1} = b_{j1}, a_{i2} = b_{j2}, ...a_{ik} = b_{jk}$.

Аналогичным образом можно сформулировать задачу поиска наибольшей возрастающей подпоследовательности. Дан массив $a_1, a_2, ...a_n$. Необходимо найти такое максимальное множество индексов $i_1 < i_2 < ... < i_k < n+1$, что $a_{i1} < a_{i2} < a_{i3} < ... < a_{ik}$.

Заметим, что задачу поиска наибольшей возрастающей подпоследовательности можно свести к задаче поиска наибольшей общей подпоследовательности, если взять в качестве массива b отсортированный по возрастанию массив a.

Задача поиска наибольшей общей подпоследовательности решается методом динамического программирования. Будем хранить в динамике $dp[i][j]$ - длину наибольшей общей подпоследовательности на префиксе длины i массива a и префиксе длины j массива b. Тогда $dp[i][j] = max(dp[i-1][j], dp[i][j-1])$, если a_i не равно b_j. Если же $a_i = b_j$, то $dp[i][j] = max(dp[i-1][j], dp[i][j-1], dp[i-1][j-1]+1)$.

Аналогичным образом можно решить и задачу поиска наибольшей возрастающей подпоследовательности. Будем хранить в $dp[i]$ длину наибольшей возрастающей подпоследовательности, заканчивающейсв в i-том элементе. Тогда для вычисления $dp[i]$ переберем все возможнные варианты предедущего элемента нашей последовательности, то есть значения $j < i$ такие что $a_j < a_i$. По всем таким варинатам возьмем максимум. Данный максимум, увеличенный на 1 (приписываем последний элемент подопоследовательности a_i) и будет $dp[i]$.

Сложность задачи поиска наибольшей общей подпоследовательности $O(nm)$, сложность поиска наибольшей возрастающей подпоследовательности $O(n^2)$. Однако, задачу поиска наибольшей возрастающей подпоследовательности можно ускорить до $O(n * log(n))$, если для поиска максимума использовать дерево отрезков.