**Билет №11**

**Задача поиска наибольшей возрастающей подпоследовательности в массиве. Решение за O(N^2) методом динамического программирования.**

НВП – наибольшая возрастающая подпоследовательность

**Постановка задачи:** Дан массив чисел a длины n. Необходимо найти подпоследовательность этого массива максимальной длины такую, чтобы все числа этой подпоследовательности шли строго в возрастающем порядке.

Пример: a = [0, 15, 10, 16, 4].

[0, 10, 4] - подпоследовательность

[16] - подпоследовательность

[0, 15, 2] - не подпоследовательность

[0, 16, 15] - не подпоследовательность

Простой вариант решения: переберем все подпоследовательности, потом проверим на возрастания, возьмем максимальную длину. Однако сколько всего существует подпоследовательностей? Сколько с учетом этого количества займет проверка на возрастание?

Подпоследовательностей- (2^n - 1)

С учетом проверки на возрастание -Изображение выглядит как текст

Автоматически созданное описание - Это очень много

=> нужно найти другой способ

Жадный алгоритм – алгоритм, при котором принимаются локально оптимальные решения на каждом этапе. Пример: большие и мелкие купюры.

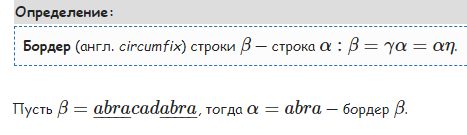
Для данной задачи не подойдет жадный алгоритм, тк например, если наибольший элемент стоит в самом начале.

Попробуем воспользоваться методом динамического программирования. По аналогии в dp[i] будем рассматривать префикс массива. Однако заметим следующее: dp[i] - НВП для префикса a от 1 до i включительно не получится, потому что неясно, как пересчитывать (неясно, как можно в наилучшую НВП для предыдующих добавлять a[i], ведь последнее значение найденной НВП в dp[i - 1] может быть больше a[i]) В общем, нужно еще наложить какое-нибудь условие

Поступим так: пусть dp[i] - НВП префикса а от 1 до i такая, что она заканчивается ровно в i-ой позиции.

ДП — термин, который лично у меня никаких правильных ассоциаций не вызывает, я бы назвал этот подход так — “Программирование с сохранением промежуточного результата этой же задачи, но меньшей размерности”.

Префикс-функция (англ. prefix-function) от строки — массив длин наибольших бордеров для каждой позиции этой строки



<https://habr.com/ru/post/343210/> - вот понятное решение