# Домашнее задание №1

Д.А. Першин

17 октября 2014 г.

## 1 Словесное описание алгоритма

При решении данной задачи будем использовать метод динамического программирования. Будем обрабатывать входной массив в обратном порядке, чтобы найти наименьшую в лексикографическом порядке подпоследоветльность. Назовем его $\alpha$. Создадим четыре массива: $L_<$, $L_>$, $P_<$, $P_>$. В этих массивах будем хранить элементы $l_{<i}$, $l_{>i}$, $p_{<i}$, $p_{>i}$ соответственно, где

$l_{<i}$ - длина наибольшей чередующейся подпоследовательности $\alpha_{x_0} \dots \alpha_{x_k}\alpha_i$ обратной входной последовательности $\alpha$, заканчивающийся в $\alpha_i$, такой, что $\alpha_{x_k} < \alpha_i$.

$l_{>i}$ - длина наибольшей чередующейся подпоследовательности $\alpha_{x_0} \dots \alpha_{x_k}\alpha_i$ обратной входной последовательности $\alpha$, заканчивающийся в $\alpha_i$, такой, что $\alpha_{x_k} > \alpha_i$.

$p_{<i}$ - индекс предыдущего элемента для $\alpha_i$ в наибольшей чередующейся подпоследовательности $\alpha_{x_0} \dots \alpha_{x_k}\alpha_i$ обратной входной последовательности $\alpha$, заканчивающийся в $\alpha_i$, такой, что $\alpha_{x_k} < \alpha_i$.

$p_{>i}$ - индекс предыдущего элемента для $\alpha_i$ в наибольшей чередующейся подпоследовательности $\alpha_{x_0} \dots \alpha_{x_k}\alpha_i$ обратной входной последовательности $\alpha$, заканчивающийся в $\alpha_i$, такой, что $\alpha_{x_k} > \alpha_i$.

**Алгоритм:**

1. Заполним массивы $L_<$ и $L_>$ значением 1, а массивы $P_<$ и $P_>$ значением $-1$.
2. Для каждого элемента $\alpha_i$ из $\alpha$ найдем значения $l_{<i}, p_{<i}, l_{>i}, p_{>i}$, заполнив массивы $L_<$, $L_>$, $P_<$, $P_>$ следующим образом:

   - если $\alpha_j < \alpha_i$ и $L_>[j] > L_<[i]$, то $L_<[i] = L_>[j] + 1$, а $P_<[i] = j$.
   - если $\alpha_j > \alpha_i$ и $L_<[j] > L_>[i]$, то $L_>[i] = L_<[j] + 1$, а $P_>[i] = j$.

   где $j \in [0, i-1]$

3. В массивах $L_<$ и $L_>$ найдем максимум (при этом, если максимумов несколько, то выбираем максимум с наибольшим индексом(так как входной массив обрабатывается в обратном порядке), чтобы найти наименьшую в лексикографическом порядке подпоследовательность). Это будет индекс первого элемента искомой подпоследовательности.

4. $l = 0$

5. Восстановим найденную подпоследовательность с помощью массивов $P_<$ и $P_>$ следующим образом:
   пусть текущий элемент подпоследовательности $x_l = \alpha_{m_k}$, тогда:

   - если индекс $m_k$ был найден в массиве $L_>$, то $m_{k-1} = P_>[m_k]$;
   - если индекс $m_k$ был найден в массиве $L_<$, то $m_{k-1} = P_<[m_k]$;
   - если $m_{k-1} = -1$, переходим к пункту 7;
   - $l = l + 1$

6. Меняем массив поиска с $L_>$ на $L_<$ или наоборот. Далее повторяем предыдущий пункт.

7. Подпоследовательность $\alpha_0 \dots \alpha_l$ - искомая последовательность.

## 2 Доказательство корректнсти

Доказательство будем строить по индукции. Для одного элемента алгоритм работает корректно. Предположим, что для $n$ элементов алгоритм также работает корректно. Рассмотрим ситуацию, когда к последоветельности из $n$ элементов добавдяется еще один.
Заполнять элементы массивов $P_<$ и $P_>$ будем следующим образом:

$$P_>[n] = \mathop{\arg\max}_{0 \le i < n;\ \alpha_j > \alpha_i} (L_<[i])$$

$$P_<[n] = \mathop{\arg\max}_{0 \le i < n;\ \alpha_j < \alpha_i} (L_>[i])$$

при этом для выполнения условия вывода последовательности с минимальным $i_k$ будем выбирать самые левые из возможных индексов.

Заполнять элементы массивов $L_<$ и $L_>$ будем следующим образом:

$$L_>[n] = L_<[P_<[n]] + 1$$

$$L_<[n] = L_>[P_>[n]] + 1$$

Предположим, что найденная чередующейся подпоследовеьтльность не является наибольшей, но в таком случае существует другая чередующаяся подпоследовательность $\alpha_{m_0} \dots \alpha_{m_k} \alpha_n$, такая что $\alpha_{m_k} > \alpha_n$, следоветельно $L_<[m_k] > L_<[P_>[n]]$, что протеворечит ранее описаным условиям

$$P_>[n] = \underset{0 \le i < n;\ \alpha_j > \alpha_i}{\arg\max} \ (L_<[i])$$

Таким образом $\alpha_{x_0} \dots \alpha_{P_>[n]} \alpha_n$ является наибольшей чередующейся подпоследовательностью. Аналогичным образом доказывем корректность для $P_<[n]$

В итоге получаем предыдущий индекс для каждого элемента последоветльности и длины наибольших чередующеихся подпоследовательностей. Из этих данных легко восстановить наибольшую чередующуюся подпоследовательность $\alpha_{x_0} \dots \alpha_{x_k} \alpha_n$ входной последовательности.

## 3 Асимптотические оценки

В результате получаем сложность по памяти $O(n)$, так как мы используем 4 массива длиной $n$.Сложность по времени равна $O(n^2)$, так как для каждого элемента $\alpha_i$, где $0 < i < n - 1$ просматривается не более $n$ элементов для массивов $L_<$ и $L_>$, а для восстановления индексов последовательности просматривется не более $n$ элементов из массивов $P_<$ и $P_>$ и не более $n$ элементов исходного массива для восстановления самой последоветльности.