Задача 1

Пусть dp_i — количество возрастающих последовательностей, заканчивающихся в i элементе.

Важно

Считать будем так, что бы $\forall i \neq j$

$dp_i \cap dp_j = \varnothing$

Даже если $a_i = a_j$, это важно

То есть не считаем по несколько раз одни и теже подпоследовательности.

Забавно, но начальных значаний видимо нет (учтем при пересчёте)

Пересчет:

Идем по возрастанию i и считаем dp_i Пусть текущий элемент (значение) x, пересчитываем dp_i.

Тогда два варианта:

1. x встречается впервые.

   Тогда $dp_i = \sum_{j=0}^{i-1} dp_j$ Это правда, потому что все dp_j попарно не пересекаются и не будут в итоге пересекаться с dp_i, в силу того что отличаются от него в последнем элементе.

2. x встречался ранее на позиции k.

   Тогда $dp_i = \sum_{j=k}^{i-1}$

   Посмотрим почему так.

   Все последовательности, такие что $a[k+1, i-1]$ элементы не входят, уже посчитаны в dp_k, так как $a_k = x$. Потому нет смысла считать их дважды. Все последовательности включающие элементы $a[k, i-1]$ и элемент x будут уникальны и не будут пересекаться с dp_i, потому что отличаются в последнем элементе, либо имеют элемент из $a[k, i-1]$. Сумму посчитаем префиксными суммами, которые в процессе будем заполнять.

Итого, если быстро умеем находить и обновлять последнее вхождение элемента, то и динамику такую насчитаем.

Последние вхождения храним в виде массива на n элементов, каждый раз его обновляя.

Так как dp_i попарно не пересекаются, то ответ на задачу — сумма всех dp_i

Все подпоследовательности учли, потому что перебрали все возможные конечные элементы.

Задача 2 *Ку-ка-ре-ку*

Тут надо кукарекнуть что динамика НОП при строке и развернутой строке эк-

вивалентна динамике по поиску наибольшего подпалиндрома.
Как искать наибольший подпалиндром?
$dp_{i,j} =$ длина наибольшего подпалиндрома на $a[i, j]$

Тогда если $a_i = a_j$, то очевидно что $dp_{i,j} = dp_{i+1,j-1} + 2$
А если нет, то один из краёв в оптимальном ответе точно роли не играет и $dp_{i,j} = \max(dp_{i+1,j}, dp_{i,j-1})$
Начальные значения $\forall i, dp_{i,i} = 1$ (очевидно)
Пересчет в порядке увеличения величины $j - i$ (длины отрезка)
Как доказать узнаем на разборе.

Задача 3 (Всем извесная *баян-задача с иннополиса :)*)
Насчитаем несколько штук.
$\mathrm{d1}, \mathrm{d2}, \mathrm{cnt1}, \mathrm{cnt2}$
d1 — длина НВП на $\{a_0, a_1, \ldots, a_i\}$, если a_i точно берём в ответ.
d2 — тоже самое на $\{a_i, a_{i+1}, \ldots, a_{n-1}\}$
cnt1 — количество НВП на $\{a_0, a_1, \ldots, a_i\}$, если a_i точно входит в НВП.
cnt2 — количество НВП на $\{a_i, a_{i+1}, \ldots, a_{n-1}\}$, если a_i точно входит в НВП.
Фух, погнали писать
$\mathrm{d1}, \mathrm{d2}$ мы считать умеем (научились на парах).
cnt1 сейчас считать научимся, а cnt2 считается аналогично первому.

Вспомним как мы считали динамику для НВП за квадрат.
Нужно на префиксе найти элемент меньший нашего с максимальным значением dp.
Утверждается что $cnt_i = \sum\limits_{j|a_j<a_i, dp_j \to max} cnt_j$
Как такое посчитать? Хороший вопрос.
Будем использовать дерево отрезков, которое мы ещё к сожалению не прошли (завуалированное под разделяй и властвуй, которое уже прошли).

Храним в доп. массиве b по значению элемента (a_i), максимальную dp, заканчивающуюся на a_i и сумму cnt по таким dp
Теперь нам нужно найти на префиксе этого массива максимальный элемент по значению dp и сумму cnt по таким элементам.

Построим разделяй и властвуй и закэшируем ответы для каждого отрезка, который посетили. (Тут инструкция как ДО пишется).
Когда говорят обновить элемент — изменятся не более чем $\log_2 n$ отрезков. Пересчитаем значания начиная от листа и заканчивая корнем.
Когда поступает запрос на отрезке найти минимум и сумму — жадно разобьем отрезки на подряд идущие степени двойки от старшей в начале отрезка (в нуле)

к младшей в конце отрезка.
Легко заметить что все такие отрезки у нас есть сохранённые.

Как объединять ответ для двух отрезков?
Просто смотрим на максимумы
Равны? Тогда ответ — максимум dp и сумма cnt_l, cnt_r
Не равны? Тогда ответ — один из максимумов по dp и соответствующая cnt

Отлично, посчитали необходимые штуки. Пусть НВП всего массива имеет длину L и количество таких подпоследовательностей всего K
Проходим каждый элемент.
Если $\mathrm{d1}_i + \mathrm{d2} - 1 < L$, то элемент не входит ни в одну НВП, потому что если его обязательно взять, то макс НВП не выйдет, как видно из неравенства. Иначе если $\mathrm{cnt1}_i \cdot \mathrm{cnt2} < K$, то элемент лежит хотя бы в одной НВП, но не лежит во всех НВП сразу. Ну эта формула есть ровно количество НВП, при условии что элемент i мы точно взяли (привет, комбинаторика)
Иначе третий вариант методом исключения. В итоге посчитали за $\mathcal{O}(n \log n)$ хорактеристику каждого элемента.