Уборка листьев

Автор задачи: Мария Жогова, разработчик: Владислав Власов

Отсортируем массив и посчитаем на нём префиксные суммы $p_i = a_1 + a_2 + \ldots + a_i$. Для этого достаточно сделать $p_0 = 0$ и посчитать в цикле каждый p_i как $p_{i-1} + a_i$. Так мы сможем узнавать сумму на любом отрезке массива a за $\mathcal{O}(1)$: $a_i + a_{i+1} + \ldots + a_j = p_j - p_{i-1}$.

Заметим, что достаточно перебрать $l = 1$ и $l = a_i + 1$ для всех возможных i. Действительно, если оптимален некоторый отрезок $[l, r]$ с другим l, его можно сдвинуть влево до ближайшего из указанных значений l, при чем слева в отрезок не войдут никакие новые a_i (потому что двигаемся до ближайшего $a_i + 1$), а справа точно ничего не добавится в отрезок, но возможно еще что-то даже выйдет за его рамки и сделает ответ лучше.

Таким образом, рассмотрим в качестве l единицу и все возможные $a_i + 1$, пока $r = a_i + k \leqslant c$, и выберем лучшую сумму. Находить для каждых l и r, какие a_i попадут в отрезок, можно либо двоичным поиском, либо методом двух указателей. Сумму на найденном отрезке можно будет найти с помощью префиксных сумм, посчитанных ранее.

Рассмотрим решение методом двух указателей: будем перебирать a_i в порядке возрастания для левой границы и искать последнее a_j, такое что $a_j \leqslant a_i + k$. Так как $a_i \leqslant a_{i+1}$, j для каждого следующего i будет не меньше, чем для $i - 1$, поэтому просто начнем перебирать j, начиная с найденного на предыдущей итерации значения. Поскольку j проходит по элементам массива, оно увеличится не больше n раз, и время работы такого решения будет $\mathcal{O}(n)$ плюс $\mathcal{O}(n \log n)$ на сортировку в начале.