Покупка на подаръци

Подзадача 1

Для решения первой подзадачи достаточно реализовать полный перебор действий Саши. В каждом отделе торгового центра у Саши есть два возможных варианта действий, значит всего вариантов его действий порядка $\mathcal{O}(2^n)$. Для каждого варианта необходимо найти стоимость самого дорогого подарка, купленного среди первых магазинов и стоимость самого дорогого подарка, купленного среди вторых магазинов, и вычислить модуль разности их стоимостей. Следует не забыть проверить, что каждая из подруг получит хотя бы один подарок.

Время работы: $\mathcal{O}(2^n \cdot n)$.

Подзадача 2

Для решения второй подзадачи зафиксируем номер отдела i, в котором будет куплен самый дорогой подарок для первой подруги, а также номер отдела j $(i \neq j)$, в котором будет куплен самый дорогой подарок для второй подруги. Осталось проверить, можно ли купить подарки в остальных отделах таким образом, чтобы подарок a_i был самым дорогим подарком, купленным для первой подруги, а подарок b_j — самым дорогим подарком, купленным для второй подруги.

Научимся выполнять данную проверку за $\mathcal{O}(n)$. Рассмотрим некоторый отдел k $(k \neq i,\ k \neq j)$. В данном отделе Саша может либо купить подарок для первой подруги стоимостью a_k, либо купить подарок для второй подруги стоимостью b_k. Если $a_k \leq a_i$, то Саша может купить в данном отделе подарок для первой подруги, и подарок a_i все еще будет самым дорогим. Аналогично, если $b_k \leq b_j$, то Саша может купить в данном отделе подарок для второй подруги. В противном случае, Саша не сможет купить подарок в данном отделе таким образом, чтобы подарки a_i и b_j были самыми дорогими, и выбранная пара (i, j) не подходит. Осталось среди всех подходящих пар (i, j) выбрать оптимальную.

Время работы: $\mathcal{O}(n^3)$.

Подзадача 3

Для начала отсортируем все отделы по убыванию b_i (а при равенстве — по возрастанию a_i). Теперь переберем отдел i, в котором будет куплен самый дорогой подарок для второй подруги. Заметим, что во всех отделах с номерами $j < i$, Саша должен купить подарок для первой подруги, иначе подарок i не будет обладать максимальной стоимостью среди подарков, купленных для второй подруги. Поэтому сразу найдем значение $m = \max\limits_{j<i} a_j$. Таким образом, мы уже можем получить ответ $|m - b_i|$.

Во всех отделах с номерами $j > i$, для которых $a_j \leq m$, Саша может купить подарок для любой из подруг, и это никак не повлияет на ответ. Теперь рассмотрим все отделы с номерами $j > i$, для которых $a_j > m$. Если в некотором из подобных отделов купить подарок для первой подруги, то значение m увеличится, а значит ответ может улучшиться. Поэтому переберем все таким отделы и обновим ответ значением $|a_j - b_i|$.

Время работы: $\mathcal{O}(n^2)$.

Подзадача 4

Отсортируем все отделы по убыванию a_i (а значит, и b_i тоже, так как $a_i = b_i$). Заметим, что в первом отделе Саша купит подарок одной из подруг, стоимость которого максимальна среди всех доступных подарков. Также заметим, что оптимальным решением является покупка подарка второй подруге во втором отделе, потому что любой другой вариант не увеличит стоимость ее подарка, а значит разность стоимостей подарков не уменьшится. Таким образом, ответом является величина $a_1 - a_2$.

Время работы: $\mathcal{O}(n \log n)$.

Подзадача 5

Возьмем решение для подзадачи 3 и оптимизируем его до $\mathcal{O}(n \log n)$. Для начала вместо того, чтобы вычислять величину m на каждой итерации заново, будем поддерживать ее значение в некоторой переменной. Тогда при переходе от отдела $i-1$ к отделу i будем обновлять значение m следующим образом: $m := \max(m, a_i)$.

Осталось научиться быстро находить оптимальный номер отдела j, такой что $j > i$, $a_j > m$, а также $|a_j - b_i|$ минимально. Выберем на суффиксе массива минимальное a_j, такое что $a_j \geq b_i$, а также максимальное a_j, такое что $a_j \leq b_i$. Можно заметить, что оптимальным a_j является одно из двух выбранных чисел (нужно также не забыть проверить условие $a_j > m$). Поэтому, достаточно обновить ответ только при помощи них.

Искать данные два элемента можно, используя структуру данных `set`. Будем поддерживать в множестве все a_j, находящиеся на суффиксе. Тогда можно найти нужные два элемента за $\mathcal{O}(\log n)$. При переходе от отдела $i-1$ к отделу i нужно удалить значение a_{i-1} из структуры данных.

Время работы: $\mathcal{O}(n \log n)$.