

## 1 Обязательные задачи

1. Если все клозы содержат больше одной переменной, то просто выдадим всем переменным true и получим успех. Иначе выполним все клозы, состоящие только из одного отрицательного литерала – если после этого какой-то из оставшихся будет невыполним, то мы проиграли, иначе снова выдадим всем оставшимся true. Работает за линию.
2. Отсортируем числа по возрастанию и заведём переменную  $p = 0$ , которая значит "какой префикс умеем набирать". Переберём числа. Пусть текущее  $= b$ . Если  $b > p + 1$ , то ответ  $= p + 1$ . Иначе  $p += b$ . В конце если нигде не сломались, то ответ  $p + 1$ .
3.
  - а) Если веса на рёбрах, то XOR пути  $a-b$  равен XOR'у путей ROOT-а и ROOT-б  $\rightarrow$  обойдём дерево dfs-ом и будем пихать в хэшмап ксоры на путях от корня до вершины и проверять, есть ли в хэшмапе путь с ксором  $S \oplus \text{cur}$ , где  $\text{cur}$  – текущий ксор. Работает за линию вообще.
  - б) Делаем центроидную декомпозицию  $\rightarrow$  задачу можно заменить на "найти проходящий через корень путь"  $\rightarrow$  тоже пихаем величины в хэшмап. Чтобы памяти было  $O(n)$ , достаточно после каждого обхода центроидной компоненты чистить этот хэшмап.
4.
  - б) Делаем центроиду, сводим задачу на "сколько проходящих через корень путей". Пусть сейчас мы обходим какое-то поддереву корня и находимся в вершине на глубине  $d$ . Тогда нас интересуют все вершины из предыдущих поддеревьев корня на глубинах от  $L - d$  до  $R - d$ . Давайте заведём специальный массив  $\text{cnt}[i]$  = кол-во вершин на глубине не больше  $i$  среди всех обойдённых поддеревьев. Тогда нам нужен запрос суммы на отрезке – это одно вычитание префиксных сумм. Когда мы обойдём поддерево, этот массив надо будет обновить – причём полностью, ведь это префиксные суммы. Давайте тогда обходить поддеревья в порядке увеличения глубины, тогда после каждого мы будем проводить модификацию за  $O(\text{размер поддерева})$ .
5.
  - а) `prerecalc` – построим центроиду, запомним расстояние до каждого нашего центроидного предка. Когда красим вершину  $v$  в чёрный, то обновляем ответ для неё и всех её предков  $p$  величиной  $d[v][p]$ . Когда запрос для вершины  $v$ , то перебираем предка  $p$  и выбираем минимум среди  $\text{dist}[v][p] + \text{ans}[p]$ .
  - б) Теперь надо ещё и красить вершины обратно – обновлением ответа для предка будет не просто релаксация величиной  $d[v][p]$ , а запишем эту величину в кучу, а потом будем её лениво удалять, если нас покрасили обратно.

## 2 Дополнительные задачи

1.

2.

3.

а) Построим MST.

б) Найдём кратчайший путь.

в) В любом дереве, содержащего эти вершины, найдётся одна такая, что пути в дереве от неё до этих трёх сойдутся в ней  $\rightarrow$  запустим три Дейкстры и найдём  $v$  с минимальной величиной  $d[a][v] + d[b][v] + d[c][v]$ .