

1 Обязательные задачи

- 1.
2. Идём слева направо, держим ДД уже использованных ключей, изначально оно пустое. Число i вставляем на позицию k_i . a_i = позиции числа i в самом конце.
3. Держим корневую по кускам, каждый кусок – ДД. Ещё для каждого куска держим его текущий inc . get – не больше корня запросов “сколько чисел меньше inc в ДД” не больше корня запросов “умма чисел больших inc ” по хвостам проходимся глупо. inc – это для каждого целого куска повесить его inc , а для хвостов честно поменять значения в ДД. Итого корень на лог на запрос.
4. Бинарный поиск по ответу + dynamic connectivity. Корень на лог.
5.
 - а) Можно за $O(\text{LCA})$ сравнением tin -ов и высот, но если центроида, то делаем персистентное ДД для каждого пути от центроиды до вершины. Соответственно можем безболезненно добавлять вершину в ДД её предка. Ответ на запрос – найти центроиду и сделать $\text{find}(v)$ в двух ДД.
 - б) Корневая по запросам. В каждом куске меняется вес не больше корня рёбер. На компонентах связности всех остальных построим центроиды. Ответ на запрос – если вершины лежат в одной компоненте, то это запрос в центроиду, иначе выписываем путь по компонентам между ними и спрашиваем у каждой центроиды + ещё обрабатываем те особые рёбра, которые меняются в этом куске. Корень на лог.

2 Дополнительные задачи

- 1.