

Задание 1

Возьмите алгоритм подсчёта массива ID с лекции и сделайте так, чтобы он использовал $O(n^{1+\varepsilon})$ ($0 < \varepsilon < 1$) памяти вместо $O(n^2)$, сохранив асимптотику work и span, если ε - константа. Подсказка: время работы изменится где-то в $\frac{1}{\varepsilon}$ раз.

1. Вместо хранения полной таблицы BB, на шаге итерации it формируем ключи (старый ранг, ранг смещённой позиции и исходный индекс):

$$K_i = (ID[it][i], ID[it][i + 2^{it}], i), \quad i \in [0, n)$$

2. Отсортировать массив всех ключей по первой и второй компоненте стабильной сортировкой
 - используем поразрядную сортировку в системе счисления с основанием n^ε
 - реализуем в нескольких `parallel_for`: подсчёт, префикс-сумма, распределение
3. Память:
 - два массива рангов ID длины n (текущий и следующий)
 - массив пар P длины n
 - временный массив для перестановки длины n
 - массив счётчиков длины n^ε
4. После сортировки одинаковые пары будут стоять подряд, и одному блоку подряд идущих элементов можно присвоить один новый ранг $ID[it + 1]$