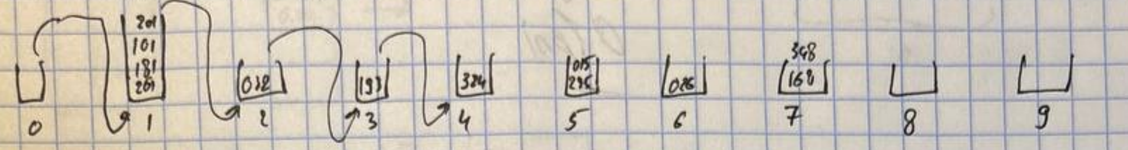
Если сортировать не со старшего разряда, а с младшего. (LSD)

261 324 181 168 295 348 101 015 026 032 193 201



$O(n)$

Выписываем массив в таком порядке

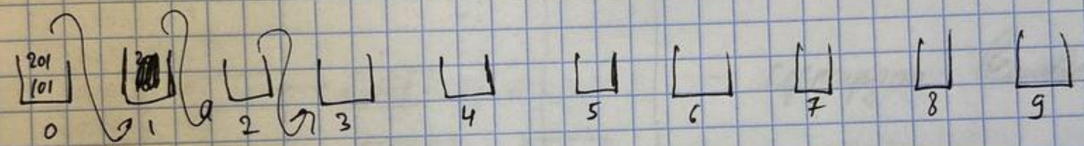
261 181 101 201 032 193 324 235 015 026 168 342

Нужно склеить связные списки, а это $O(1)$.

Нужно склеить r списков $\Rightarrow O(r)$.

Итого $O(n+r)$

Раскладываем по 2 разряду



Получаем

101 201 015 324 026 032 348 261 168 181 193 295

Числа упорядочены по последним двум разрядам.

Нужно в последний раз сделать и они отсортируются целиком.

Итого $O(\underbrace{p}_{\text{число разрядов}}(n+r)) = O(pn + pr)$

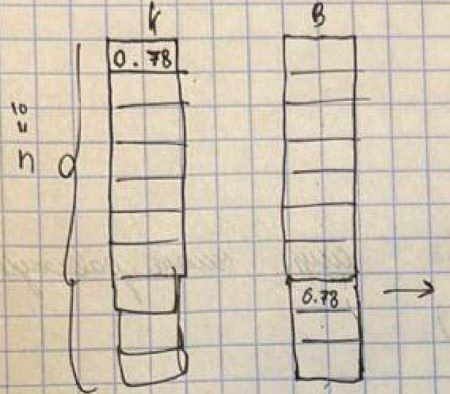
$p = \log_r n$, $r = const$; $p - const$

$O(n \log n + \log n)$ $\quad$ $O(n)$

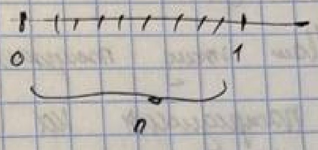
Карманная сортировка

$K[i] \in [0,1)$ $\qquad$ $B[\lfloor n \cdot K[i] \rfloor] = K[i]$

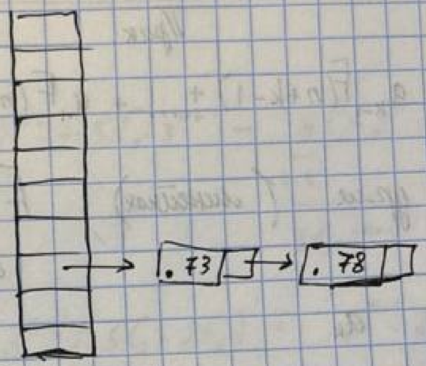
Форм. массив по такому правилу.

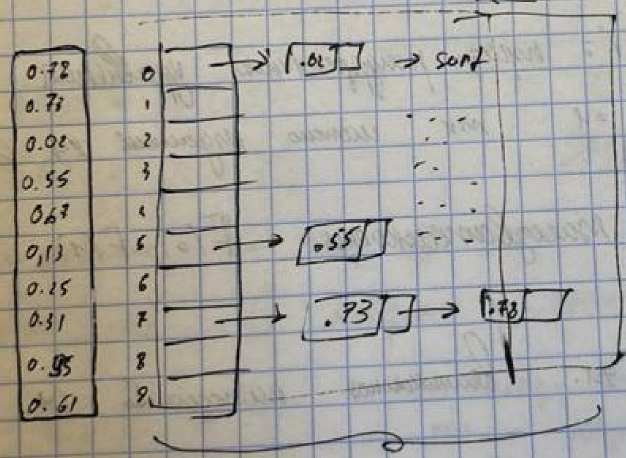


Делим интервал от 0 до 1 на n частей. Находим номер соответствующего отрезка.



массив указателей на связные списки





n раз по $O(1)$

сортируем вставкой

$O(n)$

$O(n \log n)$ ← если очень надо (можно, но не очень нужно)

- В лучшем случае числа равномерно распределены по в каждом кармашке по 0 или 1 эл. => сорт. $O(n)$

  Тк сорт вставк $O(1)$

- В худшем случае все попадут в один кармашек

  => $O(n + n^2) = O(n^2)$

• В среднем случае $T(n) = O(n) + \sum_{i=0}^{n-1} O(n_i^2)$

$$M[T(n)] = O(n) + \sum_{i=0}^{n-1} O(\underbrace{M[n_i^2]}_{2-\frac{1}{n}}) =$$

$$= O(n) + n\cdot\left(2-\frac{1}{n}\right) =$$

$$= O(n) + 2n - 1 = O(n)$$

Мат. ожид. построено на том, что случ. числа равномерно распределены на интервале $[0,1)$

Прак.

$$a_k F(n+k) + a_{k-1} F(n+k-1) + \dots + a_1 F(n+1) + a_0 F(n) = f(n)$$

Рекуррентное ур-ие (линейное); F — числовая посл-ть, аргументы — целые неотр. числа

Поделим на a_k

(*) $F(n+k) = \dots$; k: порядок рекуррентного уравнения

будем считать, что коэфф $=1$ т.к. можно поделить на a_k

• Решение рекурр. ур-ия — последовательность $F(0), F(1), \dots F(i)$ она рекурсивна

· Для любых членов ст. эл. выполняется тождество (*)

• Решение (замкнутый вид): формула для i-ого элемента $F(i)$

Порядковые статистики

n элементов

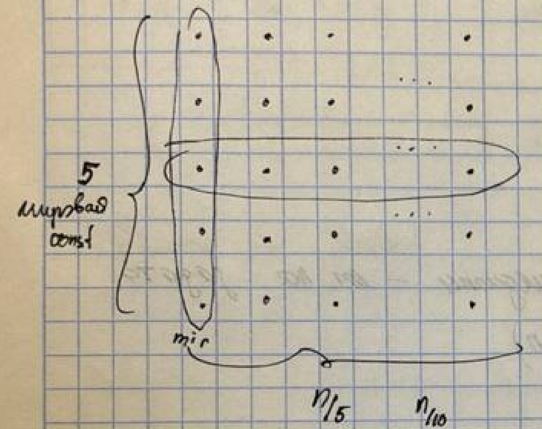
$k : n$ — k-тая порядковая статистика

$1 : n$ — min $\qquad \frac{n}{2} : n$ — медиана

$n : n$ — max

Вычислить k-ую порядковую статистику.

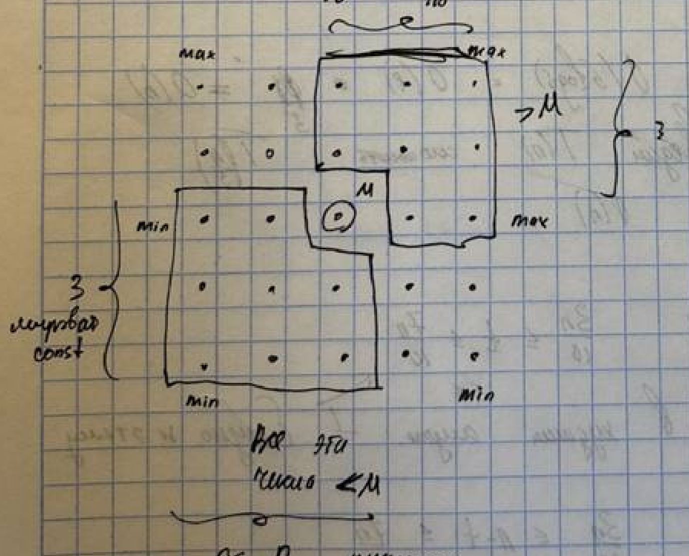
Нужно сделать сортировку $(n \log n)$

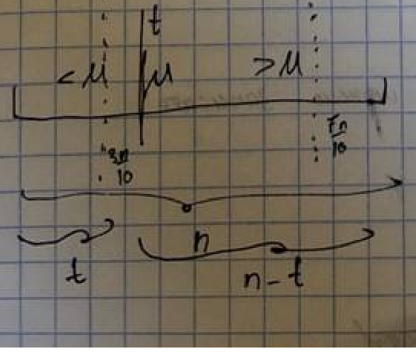


сортируем все столбцы

найти медиану медиан

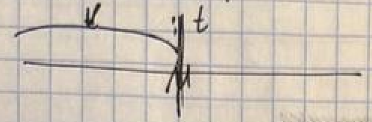
медиана медиан окажется в центре таблицы





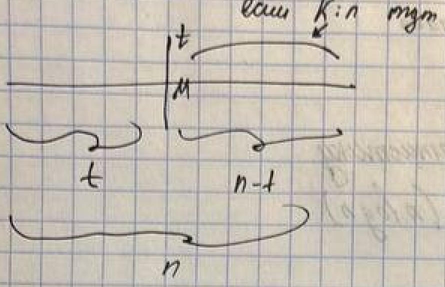
partition сделали разбиение по t

примерно $\frac{3n}{10}$ слева и $\frac{7n}{10}$ справа если k-ая пор. статистика здесь



$k < t: \quad k:n \Rightarrow k:t$

массив t меньше и мы пришли к рекурсии



$k > t: \quad k-t : n-t$

Откуда возьмем M?

нахождение медианы – эта же задача $\left(\frac{n}{2} : n\right)$

Сводим вместе:

1) Сортировка столбцов $\quad O(5\log 5) = O(1) \cdot \frac{n}{5} = O(n)$
2) Поиск M $\quad$ Введем $T(n)$ сложность $T\left(\frac{n}{5}\right)$
3) partition дает t $\quad O(n)$
4) $k < t \Rightarrow k:t$
   $k \geq t \Rightarrow k-t : n-t$

$$T\left(\frac{7n}{10}\right) \begin{cases} \frac{3n}{10} \le t \le \frac{7n}{10} \\ \text{в худшем случае } t \text{ близко к этому} \\ \frac{3n}{10} \le n-t \le \frac{7n}{10} \end{cases}$$

$$T(n) = T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right) + \underbrace{O(n)}_{2n}$$

представим как $c \cdot n$

можно как $n\log n$, но это бессмысленно.

можно записать

$$c \cdot n = \dots \; c \cdot n + O(c \cdot n)$$

должна быть эта же const

Предп что c – конкр. const

$$10n = \frac{10n}{5} + \frac{10n}{10} \cdot 7 + 2n = 11n$$ const не совпали.

$$20n = \frac{20n}{5} + \frac{20n}{10} \cdot 7 + 2n = 20n$$

$$T(n) = 20n$$

В итоге, порядковую статистику ядра можно найти с линейной сложностью.

$1 : n$ $2 : n$ $3 : n$ можем отсортировать массив $20 \cdot n \cdot n = O(n^2)$