

Это можно сделать в один проход по массиву a . Каждый раз, когда мы встречаем элемент массива b , мы записываем его и его номер в специальные массивы. При этом мы поддерживаем в этих массивах отрезок I , на котором мы надеемся найти все различные элементы b . Ясно, что если очередной элемент массива a совпадает с первым элементом отрезка I , то I уже явно не может быть кратчайшим отрезком, удовлетворяющим условию задачи, и мы можем сдвинуть его левый конец. Если на очередном шаге мы понимаем, что I содержит все различные элементы b , то I — кандидат на ответ; в этом случае мы также сдвигаем его левый конец.

Оценка $O(n)$ по памяти очевидна. Оценка $O(nk)$ по сложности может быть обоснована следующим образом: мы все делаем в один проход (отсюда n) и на каждом шаге должны искать элемент в массиве b (отсюда k). Ясно, что алгоритм можно улучшить: если вначале отсортировать b и использовать двоичный поиск, получим $O(n \log k)$. Если же использовать совершенное хеширование, то можно добиться сложности $O(n + k)$.