

# Алгоритмы и структуры данных.

## Непростые задачи на простые структуры, решаем вместе.

1. Предложите способ реализовать во внешней памяти структуры стек, очередь и дек, чтобы время выполнения всех операций составляло  $O(\frac{1}{B})$ .
2. Докажите, что если у вас имеется корректная куча на минимум, затем какой-то элемент уменьшается, то после применения к этому элементу операции просеивания вверх у вас будет корректная куча.
3. Докажите, что если у вас имеется корректная куча на минимум, и условие кучи выполнено для всех вершин, кроме корня, то применение к корню операции просеивания вниз создаст корректную кучу.
4. Пусть в алгоритме дека с минимумом через три стека используется потенциальная функция  $F(H) = \max(0, f(H) - s(H))$ , где  $f(H)$  — размер первого стека, а  $s(H)$  — размер второго стека. Получается ли с такой потенциальной функцией показать амортизированную оценку выполнения всех операций  $O(1)$ ?
5. **Напишите псевдокод** процедуры выполняющей объединение двух двусвязных закольцованных списков в один. Алгоритм получает два указателя на какой-то элемент каждого из списков (каждый из указателей также может иметь значение *NIL*). Гарантируется, что входные данные корректны.
6. Имеются две двоичные кучи размера  $n$ , записанные в массивах  $a[i]$  и  $b[j]$  соответственно. Оба массива индексируются с единицы. Процедура  $joinHeaps(n, a, b)$  выполняет объединение этих двух куч следующим образом:

```
void joinHeaps(int n, int* a, int* b) {
    for (int i = 1; i <= n; i++) {
        a[n + i] = b[i];
        siftUp(a, n + i);
    }
}
```

В коде выше процедура  $siftUp(a, x)$  выполняет стандартное просеивание вверх элемента  $x$  кучи, записанной в массиве  $a$ . Укажите асимптотическую зависимость от  $n$  времени выполнения процедуры  $joinHeaps$  и продемонстрируйте её неумлучшаемость.

7. Пусть в массиве  $a_i$  записаны  $n$  элементов. Для построения  $k$ -ичной кучи выполняется проход от последнего к первому элементу массива и к каждому применяется операция  $siftDown(i)$ . Докажите, что во-первых после окончания данного процесса в массиве  $a_i$  будет записана корректная пирамида, а во-вторых время работы будет  $O(n)$  для любого  $k \leq n$ .

8. Предложите структуру данных «мультимножество с операцией доступа к  $k$ -й порядковой статистике». Параметр  $k$  **не зависит от запроса** и определяется в начале работы программы. Структура данных должна поддерживать следующие операции: добавить элемент к множеству, вернуть  $k$ -ю порядковую статистику, удалить  $k$ -й элемент (при равенстве любой). При последовательных  $n$  операциях **гарантированная real-time** сложность вставки и удаления должна быть  $O(\log n)$ , сложность поиска  $k$ -й порядковой статистики должна быть  $O(1)$ . Дополнительно известно, что в любой момент времени структура будет содержать не более  $m$  элементов. Вам доступен один массив размера  $m$  для их хранения, разрешается использовать не более  $O(1)$  дополнительной памяти.
9. В массиве, к которому имеется read-only доступ записаны  $n + 1$  чисел от 1 до  $n$ . Требуется найти любое число, которое встречается хотя бы два раза за время  $O(n)$ , используя  $O(1)$  дополнительной памяти.