

# АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ ДЛЯ СТУДЕНТОВ

## Базовые структуры данных.

1. Реализуйте очередь, используя только стеки. Реализуйте операции нахождения  $\min/\max/\text{sum}/\text{gcd}$ . Амортизированное время  $O(1)$ .
- 2.\* Реализуйте дек, используя только стеки. Реализуйте операции нахождения  $\min/\max/\text{sum}/\text{gcd}$ . Амортизированное время  $O(1)$ .
3. Используя стек, научитесь вычислять выражения в постфиксной записи (это когда оператор ставится после аргументов, например, выражение  $4 - ((1 + 2) * 3)$  в постфиксной записи выглядит так:  $4\ 1\ 2\ +\ 3\ *\ -$ ).
4. Научитесь вычислять выражения в инфиксной записи со скобками (обычные выражения). Для простоты можно считать, что в выражении проставлены все скобки (то есть внутри скобок вычисляется только один оператор, например так можно:  $(4 - ((1 + 2) * 3))$ , а так — нет:  $(1 + 2 + 3)$ ).
5. Научитесь по выражению в инфиксной записи строить выражение в постфиксной записи.
6. Был массив чисел, все соседние числа были различны. С этим массивом несколько раз проделали операцию: вставить в массив два одинаковых числа рядом. По конечному состоянию массива восстановите его исходное состояние. Время  $O(n)$ .
7. Битовый счетчик хранит число в виде массива двоичных цифр. Изначально все цифры равны 0. Операция *inc* увеличивает счетчик на 1. Реализуйте операцию *inc* и докажите, что амортизированное время ее работы  $O(1)$ .
8. Мультистек — это последовательность стеков, максимальный размер  $i$ -го стека равен  $2^i$ . Изначально все стеки пустые. Новый элемент добавляется в первый стек. Если он уже заполнен, то все его элементы перекладываются во второй, если он также заполнен, то его элементы перекладываются в третий, и т.д. Покажите, что амортизированное время добавления элемента в мультистек  $O(\log n)$ .
9. Покажите, как сделать, чтобы в мультистеке стеки были всегда отсортированы (при этом амортизированное время добавления элемента все еще  $O(\log n)$ ).
10. Покажите, как в полученной структуре найти и удалить минимальный элемент за  $O(\log n)$ . (В результате должна получиться структура данных на базе стеков, в которую можно добавлять новые элементы и удалять минимум за  $O(\log n)$ ).