

Стек, очередь, списки. Задачи

Булгаков Илья, Гусев Илья

Московский физико-технический институт

Москва, 2023

Содержание

1 Задачи

Задача 1

Массив a_1, a_2, \dots, a_n обладает таким свойством: существует некоторый индекс i , такой что $a_1 < a_2 < \dots < a_i$, но $a_i > a_{i+1} > \dots > a_n$. Найдите такое i за $O(\log n)$ (массив уже расположен в памяти).

Задача 1 (Решение)

Массив a_1, a_2, \dots, a_n обладает таким свойством: существует некоторый индекс i , такой что $a_1 < a_2 < \dots < a_i$, но $a_i > a_{i+1} > \dots > a_n$. Найдите такое i за $O(\log n)$ (массив уже расположен в памяти).

Решение:

- Будем смотреть не на сами элементы, а на разности соседних
- Воспользуемся бинарным поиском

Задача 2

Как отсортировать связный список длины n за $O(n \log n)$ времени, используя $O(1)$ дополнительной памяти?

Задача 2 (Решение)

Как отсортировать связный список длины n за $O(n \log n)$ времени, используя $O(1)$ дополнительной памяти?

Решение:

- Воспользуемся нерекурсивным MergeSort:
- Отсортируем сначала блоки по 2 элемента, затем по 4, и т.д.

Задача 3

Предложите метод хранения минимального значения в очереди с помощью структуры `deque`. Если к очереди поступило n запросов, время работы программы должно составлять $O(n)$.

Задача 3 (Решение)

Предложите метод хранения минимального значения в очереди с помощью структуры `deque`. Если к очереди поступило n запросов, время работы программы должно составлять $O(n)$.

Решение:

- Когда приходит запрос на добавление в очередь числа x , убираем с конца дека все элементы большие, чем x
- Когда вынимаем элемент из очереди, проверяем, лежит ли он в начале дека. Если да, то убираем его и из дека
- Минимальное значение в очереди - первый элемент дека

Задача 4

Проверьте данную строку на то, является ли она корректной XML-строкой (то есть имеет вид `<a><uv></uv>`).

Задача 4 (Решение)

Проверьте данную строку на то, является ли она корректной XML-строкой (то есть имеет вид `<a><uv></uv>`).

Решение:

- Разобьем строку на токены (вида `<a>`, `` и т.д.)
- Будем поддерживать стек вложенности:
- Если обрабатываем открывающийся токен, то кладем его на стек.
- Если закрывающийся - проверяем, что на вершине стека токен нужного типа
- Проверим, что не было *Stack Underflow*, и что в конце стек остался пустым. Если это так - строка является корректной

Задача 5

Дан односвязный список с указателем на голову. Возможно, он зациклен. Определите, зациклен ли он, за $O(n)$ времени (где n — число вершин в списке) и с привлечением $O(1)$ дополнительной памяти.

Задача 5 (Решение)

Дан односвязный список с указателем на голову. Возможно, он зациклен. Определите, зациклен ли он, за $O(n)$ времени (где n — число вершин в списке) и с привлечением $O(1)$ дополнительной памяти.

Решение:

- Пустим два указателя. Один будет двигаться на одну вершину вперед, другой — на две
- Если указатели встретились, то список зациклен

Бонусная задача

Leetcode: 232. Implement Queue using Stacks

Реализуйте очередь на стеках.