

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное образовательное учреждение высшего образования
САНКТ – ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

старший преподаватель		С. Ю. Гуков
_____ должность, уч. степень, звание	_____ подпись, дата	_____ инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6

Построение кучи

по курсу: АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №	4321		К.А. Лебедев
		_____ подпись, дата	_____ инициалы, фамилия

Санкт-Петербург, 2024

СОДЕРЖАНИЕ

1 Цель работы.....	3
2 Задание.....	4
3 Ход разработки	5
4 Исходный код программы	6
5 Результаты работы программы	9
6 Вывод.....	10

1 Цель работы

Переставить элементы заданного массива чисел так, чтобы он удовлетворял свойству мин-куч.

2 Задание

Программа может быть написана на любом языке программирования. Вход. Массив чисел $A[0 \dots n-1]$. Выход. Переставить элементы массива так, чтобы выполнялись неравенства $A[i] \leq A[2i + 1]$ и $A[i] \leq A[2i + 2]$ для всех i .

Построение кучи – ключевой шаг алгоритма сортировки кучей. Данный алгоритм имеет время работы $O(n \log n)$ в худшем случае в отличие от алгоритма быстрой сортировки, который гарантирует такую оценку только в среднем случае. Алгоритм быстрой сортировки чаще используют на практике, поскольку в большинстве случаев он работает быстрее, но алгоритм сортировки кучей используется для внешней сортировки данных, когда необходимо отсортировать данные огромного размера, не помещающиеся в память компьютера. Чтобы превратить данный массив в кучу, необходимо произвести несколько обменов его элементов. Обменом мы называем базовую операцию, которая меняет местами элементы $A[i]$ и $A[j]$. Ваша цель в данной лабораторной работе – преобразовать заданный массив в кучу за линейное количество обменов. Формат входа.

Первая строка содержит число n . Следующая строка задаёт массив чисел $A[0], \dots, A[n-1]$. Формат выхода. Первая строка выхода должна содержать число обменов m , которое должно удовлетворять неравенству $0 \leq m \leq 4n$. Каждая из последующих m строк должна задавать обмен двух элементов массива A . Каждый обмен задаётся парой различных индексов $0 \leq i \neq j \leq n-1$.

После применения всех обменов в указанном порядке массив должен превратиться в мин-кучу, то есть для всех $0 \leq i \leq n-1$ должны выполняться следующие два условия: • если $2i + 1 \leq n-1$, то $A[i] < A[2i + 1]$. • если $2i + 2 \leq n-1$, то $A[i] < A[2i + 2]$. Ограничения. $1 \leq n \leq 10^5$; $0 \leq A[i] \leq 10^9$ для всех $0 \leq i \leq n-1$; все $A[i]$ попарно различны; $i \neq j$.

3 Ход разработки

Были реализованы класс задачи и функция для обработки и парсинга массив в соответствии с условиями, написана утилита для прочтения чисел из командной строки на языке TypeScript, интегрирован компилятор в JavaScript, который сразу собирает TypeScript проект без предварительной компиляции.

4 Исходный код программы

```
// index.ts

import Task from "../lib/Task";

new Task(); // инициализация задачи

// utils/readline.ts

import readline from "node:readline";

// readline интерфейс

export const readlineInterface = readline.createInterface({
  input: process.stdin,
  output: process.stdout,
});

// utils/heapsort.ts

const heapifyDown = <T extends number>(
  arr: T[],
  n: number,
  i: number,
  swaps: T[] | T[][]
) => {
  let smallest = i;
  let left = 2 * i + 1;
  let right = 2 * i + 2;
  if (left < n && arr[left] < arr[smallest]) {
    smallest = left;
  }
  if (right < n && arr[right] < arr[smallest]) {
    smallest = right;
  }
  if (smallest !== i) {
    swaps.push([i, smallest] as any);
    [arr[i], arr[smallest]] = [arr[smallest], arr[i]];
    heapifyDown(arr, n, smallest, swaps);
  }
};

export const heapify = <T extends number>(arr: T[]): T[][] => {
  const n = arr.length;
```

```

    let swaps: T[] = [];
    for (let i = Math.floor(n / 2) - 1; i >= 0; i--) {
        heapifyDown(arr as any, n, i, swaps);
    }
    return swaps as any;
};

// lib/Task.ts

import { heapify } from "../utils/heapSort";
import { readlineInterface } from "../utils/readline";

class Task {
    private arr: number[];
    private arrLength: number;

    constructor() {
        this.arrLength = 0; // длина массива
        this.arr = []; // массив
        this.read(); // инициализация задачи
    }

    // получения ответа на вопрос

    getAnswer() {
        const heap = heapify(this.arr);
        if (heap.length === 0) {
            console.log("0");
        } else {
            for (const [i, j] of heap) {
                console.log(i, j);
            }
        }
    }

    // инициализация задачи

    read() {
        readlineInterface.question("", (length: string) => {
            this.arrLength = parseInt(length); // чтение длины массива

            if (isNaN(this.arrLength) || this.arrLength <= 0) {
                console.log(
                    "Пожалуйста, введите корректное положительное число."
                );
            }
        });
    }
}

```

```

        );
        readlineInterface.close();
        return;
    }

    readlineInterface.question("", (arr: string) => {
        this.arr = arr.split(" ").map(Number);

        this.arr.length !== this.arrLength
            ? console.log("Введен не корректный массив!")
            : this.getAnswer();

        readlineInterface.close();
    });
});
}
}

export default Task;

```


5 Результаты работы программы

Результат работы программы включает в себя вывод полученной кучи

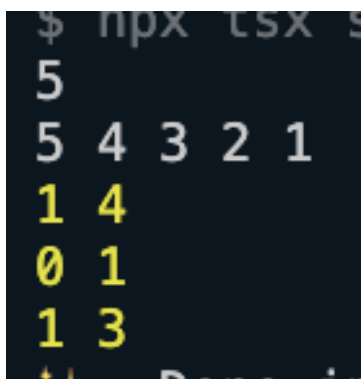


Рисунок 1 – тест программы №1

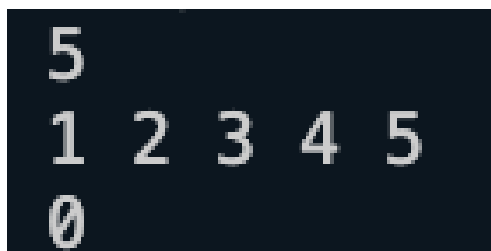


Рисунок 2 – тест программы №2

6 Вывод

Эффективное преобразование массива в мин-кучу является важной задачей в области алгоритмов и структур данных. Это преобразование позволяет организовать данные таким образом, чтобы для каждого элемента массива выполнялись определенные неравенства, что обеспечивает быструю доступность минимальных значений.

Ключевым шагом в этом процессе является выполнение обменов элементов массива, которые позволяют достичь желаемой структуры. При этом важно минимизировать количество обменов, чтобы сохранить эффективность алгоритма.

Алгоритм, который будет реализован, гарантирует, что после выполнения всех операций массив будет соответствовать свойствам мин-кучи, что открывает возможности для дальнейших операций, таких как сортировка.

Реализация данного алгоритма будет осуществляться с учетом заданных ограничений, что позволяет гарантировать корректность и эффективность работы при больших объемах данных. В результате выполнения программы будет получен не только преобразованный массив, но и информация о количестве произведенных обменов, что позволит анализировать эффективность алгоритма.