МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №5
по дисциплине «Алгоритмы и структуры данных»
Тема: Алгоритмы сортировки

Студентка гр. 9303 _____ Хафаева Н. Л.

Преподаватель _____ Филатов Ар. Ю.

Санкт-Петербург
2020

Цель работы.

Изучить такую структуру данных, как куча.

Задание.

Вариант 30. Дан массив чисел и 2 числа: n (n=1, 2, 3, …) и x. Проверить, является ли массив n-арной кучей, и если да, то, заменив корень на x, выполнить его просеивание сверху вниз.

Основные теоретические положения.

Куча — это специализированная структура данных типа дерево, которая удовлетворяет свойству кучи: если B является узлом-потомком узла A, то ключ(A) ≥ ключ(B). Из этого следует, что элемент с наибольшим ключом всегда является корневым узлом кучи, поэтому иногда такие кучи называют *max*-кучами (в качестве альтернативы, если сравнение перевернуть, то наименьший элемент будет всегда корневым узлом, такие кучи называют *min*-кучами).

Выполнение работы.

Была реализована функция *bool checker_(int* arr, int n, int i, int size),* которая принимает 4 аргумента: массив чисел, количество потомков в куче, текущий индекс и размер массива. Она каждый узел сравнивает с потомками, если узел меньше потомка функция возвращает *false*, иначе *true*. Потомки высчитываются по формуле $n * i + j$, где $j = 1, 2, 3…$

Так же была написана функция *void heap_sift(int* arr, int n, int start, int end),* которая принимает 4 аргумента: массив чисел, количество потомков в куче, начальный и конечный индексы массива. Функция выполняет просейку сверху-вниз для родителя, у которого *n* потомков. Начало подмассива - узел, с которого начинаем просейку вниз. Первый потомок (если он в пределах подмассива) назначается максимальным потомком и среди потомков

определяется максимальный. Если максимальный потомок больше родителя, то, они меняются местами, и этот потомок сам становится родителем.

Выводы.

В ходе выполнения лабораторной работы я ознакомилась с такой структурой данных, как куча.

Была реализована программа, которая определяет является ли массив кучей, заменяет узел на другое число и делает просейку сверху-вниз.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>
#include <fstream>

using namespace std;

bool checker_(int* arr, int n, int i, int size) {
	for (int j = 1; j <= n; j++) {
		if ((n * i + j) > size) {
			continue;
		}
		else {
			if (arr[i] >= arr[n * i + j]) {
				bool cur = checker_(arr, n, n * i + j, size);
				if (!cur) {
					return false;
				}
			}
			else {
				return false;
			}
		}
	}
}
void heap_sift(int* arr, int n, int start, int end) {
	int root = start;

	while (true) {
		int child = root * n + 1;
		if (child > end) {
			break;
		}

		int max = child;
		for (int i = 2; i < n + 1; i++) {
			int current = root * n + i;
			if (current > end) {
				break;
			}
			if (arr[current] > arr[max]) {
				max = current;
			}
		}
		if (arr[root] < arr[max]) {
			int cur = arr[root];
```

```
                arr[root] = arr[max];
                arr[max] = cur;
                root = max;
            }
            else {
                break;
            }
        }
}

void heap_trans(int* arr,int start, int end, int n) {
    for (int i = end; i >= start; i--) {
        heap_sift(arr, n, i, end);
    }
}

int main() {
    int* data;
    int n;
    int x;
    int size;
    string input_file;
    ifstream f;
    cout << "Enter a file name.\n";
    cin >> input_file;
    f.open(input_file);
    if (!f.is_open()) {
        cout << "Invalid file name.\n";
        return 1;
    }

    f >> n;
    f >> x;
    f >> size;
    data = new int[size];
    for (int i = 0; i < size; i++) {
        f >> data[i];
    }

    if (!(checker_(data, n, 0, size))) {
        cout << "it's not a heap";
        return 0;
    }
    else {
        if (x > data[0]) {
            for (int i = 0; i < size; i++) {
                cout << data[i] << ' ';
            }
            return 0;
        }
```

```
            else {
                  data[0] = x;
                  heap_trans(data, 0, size - 1, n);
                  for (int i = 0; i < size; i++) {
                        cout << data[i] << ' ';
                  }
            }
      }
      return 0;
}
```

ПРИЛОЖЕНИЕ Б
ТЕСТИРОВАНИЕ

	Входные данные	Результат	Комментарий
1	4 32 15 100 90 70 60 99 20 22 21 10 40 50 11 12 5 1	99 90 70 60 32 20 22 21 10 40 50 11 12 5 1	Программа работает корректно
2	2 33 5 10 20 5 6 1	it's not a heap	Программа работает корректно
3	2 100 7 55 30 35 4 10 6 8	100 30 35 4 10 6 8	Программа работает корректно