

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

ОТЧЕТ
по лабораторной работе №5
по дисциплине «АиСД»
ТЕМА: n-нарная куча

Студентка гр. 9382

Пя С.

Преподаватель

Фирсов М.А.

Санкт-Петербург

2020

Цель работы.

Познакомиться с n-нарными кучами. Научиться представлять массив в виде n-нарной кучи на примере языка C++. Выполнить работу в соответствии с заданием.

Основные теоретические положения.

Куча — это специализированная структура данных типа дерево, которая удовлетворяет свойству кучи: если B является узлом-потомком узла A, то $\text{ключ}(A) \geq \text{ключ}(B)$.

N-нарная куча – это куча с n количеством потомков. Для i-го элемента массива индексы (если отсчитывать их с нуля) его N потомков вычисляются:

1-й потомок: $N \times i + 1$

2-й потомок: $N \times i + 2$

3-й потомок: $N \times i + 3$

...

N-й потомок: $N \times i + N$

Задание

Вариант №29.

Дан массив чисел и число n ($n=1, 2, 3, \dots$). Предполагая, что массив является n-арной кучей: - Вывести его в виде n-арной кучи. - Получить путь от корня до листа такой, что при каждом шаге вниз выбирается наибольший сын.

Ход работы.

1) Разработан алгоритм:

На вход подается массив, который с помощью индексов можно представить в виде n-нарной кучи, затем задается число n. Массив выводится в виде этой кучи. Затем выводится путь от корня до листа, спускаясь вниз и выбирая наибольшего сына. Алгоритм выбирает из сыновей большего и выводит его на экран.

Предусмотрен механизм простейшего взаимодействия с пользователем, позволяющий понять алгоритм исполнения программы, с помощью вывода сообщений. Также был предусмотрен ввод данных с клавиатуры.

2) Использованы функции:

1. printHeap

Сигнатура: void printHeap(int n, int u, int q, int h).

Назначение: выводит массив в виде n-арной кучи.

Описание аргументов: n : int – количество значений в массиве, u : int – максимальное количество сыновей, q : int – количество уровней в куче, h :int – максимальная длина значения.

2. printRoot

Сигнатура: void printRoot(int n, int u).

Назначение: вывод пути от корня до листа с выбором наибольшего сына.

Описание аргументов: n: int – количество значений массива, u : int – максимальное количество сыновей.

3. fcn

Сигнатура: void fcn(std::istream &fin).

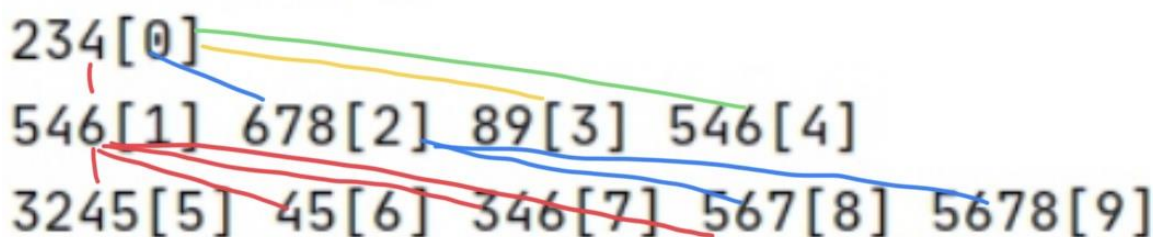
Назначение: предназначена для универсальной работы с потоками. Также в ней реализуются все остальные функции.

Описание аргументов: fin : istream – поток ввода.

Реализация вывода пути от корня до листа с выбором наибольшего сына:

На вход подаются количество значений массива и максимальное количество сыновей. Выводится корень, затем из сыновей выбирается наибольший, после сравниваются его сыновья и так далее.

Пример работы программы.



Входные данные	Выходные данные
----------------	-----------------

1 1 2 3 4 5 6 7 8 9 1 3 (было выведено со сдвигами ради наглядности)	What stream do you want to use?(0 - cout, n - name of file) 1 N-narny heap: 1[0] 2[1] 3[2] 4[3] 5[4] 6[5] 7[6] 8[7] 9[8] 1[9] Max root: 1[0] 4[3] 0[10]
--	--

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	0 6 2 24 56 4 3 4 23	N-narny heap: 8[0] 3[1] 7[2] 4[3] 6[4] 5[5] 8[6] 4[7] 3[8] Max root: 8[0] 7[2] 4[7]	Проверка на корректность работы программы с терминалом
2.	9 8 6 5 4 3 2 1 2	N-narny heap: 9[0] 8[1] 6[2] 5[3] 4[4] 3[5] 2[6] 1[7] Max root: 9[0] 8[1] 5[3] 1[7]	Проверка на корректность работы с файлом

4.	3	What stream do you want to use?(0 - cout, n - name of file) 3 Error data	Проверка на корректность работы с пустыми данными
----	---	--	---

Выводы.

В ходе работы была освоена реализация n-нарной кучи, отработано понимание его применения, и отработаны навыки письма в C++.

Код программы можно найти в приложении А.

ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <cstring>
#include <string>
#include <iostream>
#include <cmath>
#include <fstream>

#define NMAX 500
using namespace std;
int arr[NMAX+1];

/*oid heapify (int pos, int n, int u) {
    while (u * pos + 1 < n) {
        int max = u * pos + 1;
        for (int k = 2; k < u + 1; k++)
            if (u * pos + k < n && arr[u * pos + k] >= arr[max]) {
                max = u * pos + k;
            }
        if (arr[pos] < arr[max]) {
            swap(arr[pos], arr[max]);
            pos = max;
        } else
            break;
    }
}

void heapMake(int n, int u) {
    for (int i = n - 1; i >= 0; i--) {
        heapify(i, n, u);
    }
}

void heapSort(int n, int u) {
    heapMake(n, u);
    while (n > 0)
```

```

    {
        swap(arr[0],arr[n-1]);
        n--;
        heapify(0,n, u);
    }
}*/

void printHeap(int n, int u, int q, int h) { //предназначена для вывода в форме
кучи
    int k = 1, i = 0, y = 0;
    while (true) {
        /*      for (int r = 0; r < q*u*h*4 / (k + 1); r++)
            cout << " ";*/
        while (i < k + y) {
            cout << arr[i] << "[" << i << "]" "; //вывод сыновей
        /*      for (int r = 0; r < q*u*h*4 / (k + 1); r++)
            cout << " ";*/
            i++;
            if (i == n) {
                y = -1;
                break;
            }
        }
        cout << "\n";
        k *= u;
        if (y == -1) //выход из цикла
            break;
        y = i;
    }
}

void printRoot(int n, int u) { //предназначена для вывода пути от корня до листа
с выбором наибольшего сына
    int main = 0, max = 1;
    cout << arr[0] << "[0] ";
    while (true) {
        max = main + 1;
        if (max > n) {
            cout << "\n";
            return;
        }
        for (int k = 2; k < u + 1; k++)
            if (main + k < n && arr[main + k] >= arr[max]) { //выбор наибольшего
сына
                max = main + k;
            }
        cout << arr[max] << "[" << max << "]" "; //вывод сына
        main = max * u;
    }
}

void fcn(std::istream &fin) { //функция для универсальной работы с потоком ввода
    int u = 0, i = 0, value, max = 0;
    char* str = new char[30]();
    while ((str[u] = fin.get()) != '\n') { //считывание значений и запись в
массив
        if (str[u] == ' ') {
            if (max <= u)
                max = u; //максимальная длина значений
            str[u] = '\n';
            u = -1;
            value = stoi(str, nullptr, 10);
            arr[i++] = value;
            delete[] str;
        }
    }
}

```

```

        str = new char[20]();
    } else if (!isdigit(str[u])) {
        break;
    }
    u++;
}
if (!arr[0]) { //проверка на неверные данные
    cout << "Error data\n";
    return;
}
value = stoi(str, nullptr, 10); //считывание последнего элемента
arr[i++] = value;
fin >> u;
value = 0;
for (int r = 0; r < i;) {
    r = pow(u, value++);
}
cout << "N-nary heap:\n";
printHeap(i, u, value, max); //вывод в форме кучи
cout << "Max root:\n";
printRoot(i, u); //вывод пути от корня до листа с выбором наибольшего сына
delete[] str;
}

int main() {
    char n;
    cout << "What stream do you want to use?(0 - cout, n - name of file)\n";
    cin >> n;
    if (n == '0') {
        cout << "Write array of data! Write number of children!\n";
        n = cin.get();
        fcn(std::cin);
    } else {
        char *filename = new char[30](); //входные данные из файла
        strcpy(filename, "Tests//");
        strncat(filename, &n, 1);
        strcat(filename, ".txt");
        std::ifstream in(filename);
        if (!in.is_open()) {
            std::cout << "File wasn't opened!";
            return 0;
        }
        fcn(in);
        delete[] filename;
    }
    return 0;
}

```