**Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики**

**Алгоритмы и структуры данных**

**Лабораторная работа №5**

**Heap**

Выполнил:

Бараканов Жаргал Мырзабекович

Факультет ИКТ

Группа K3121

Преподаватель:

Харьковская Татьяна Александровна

**Санкт-Петербург**

**09.12.2021**

**Задание 1.**

Структуру данных «куча», или, более конкретно, «неубывающая пирамида», можно реализовать на основе массива.

Для этого должно выполняться основное свойство неубывающей пирамиды, которое заключается в том, что для каждого 1 ≤ i ≤ n выполняются условия:

* если 2i ≤ n, то ai ≤ a2i ,
* если 2i + 1 ≤ n, то ai ≤ a2i+1.

Дан массив целых чисел. Определите, является ли он неубывающей пирамидой.

* Формат входного файла (input.txt). Первая строка входного файла содержит целое число n (1 ≤ n ≤ 106). Вторая строка содержит n целых чисел, по модулю не превосходящих 2 · 109;
* Формат выходного файла (output.txt). Выведите «YES», если массив является неубывающей пирамидой, и «NO» в противном случае;
* Ограничение по времени. 2 сек.;
* Ограничение по памяти. 256 Mб.

Изображение выглядит как текст

Автоматически созданное описание

**Задание 2.**

В этой задаче ваша цель - привыкнуть к деревьям. Вам нужно будет прочитать описание дерева из входных данных, реализовать структуру данных, сохранить дерево и вычислить его высоту.

* Вам дается корневое дерево. Ваша задача - вычислить и вывести его высоту. Напомним, что высота (корневого) дерева — это максимальная глубина узла или максимальное расстояние от листа до корня. Вам дано произвольное дерево, не обязательно бинарное дерево;
* Формат ввода или входного файла (input.txt). Первая строка содержит число узлов n (1 ≤ n ≤ 105). Вторая строка содержит n целых чисел от -1 до n-1 – указание на родительский узел. Если i-ое значение равно -1, значит, что узел i - корневой, иначе это число является обозначением индекса родительского узла этого i-го узла (0 ≤ i ≤ n-1). Индексы считать с 0. Гарантируется, что дан только один корневой узел, и что входные данные представляют дерево;
* Формат вывода или выходного файла (output.txt). Выведите целое число – высоту данного дерева;
* Ограничение по времени. 3 сек.;
* Ограничение по памяти. 512 мб.

Изображение выглядит как текст

Автоматически созданное описание

**Задание 4.**

Первым шагом алгоритма HeapSort является создание пирамиды (heap) из массива, который вы хотите отсортировать.

Ваша задача - реализовать этот первый шаг и преобразовать заданный массив целых чисел в пирамиду. Вы сделаете это, применив к массиву определенное количество перестановок (swaps). Перестановка — это операция, как вы помните, при которой элементы ai и aj массива меняются местами для некоторых i и j. Вам нужно будет преобразовать массив в пирамиду, используя только O(n) перестановок. Обратите внимание, что в этой задаче вам нужно будет использовать min-heap вместо max-heap

* Формат ввода или входного файла (input.txt). Первая строка содержит целое число n (1 ≤ n ≤ 105), вторая содержит n целых чисел ai входного массива, разделенных пробелом (0 ≤ ai ≤ 109, все ai - различны.)
* Формат выходного файла (output.txt). Первая строка ответа должна содержать целое число m - количество сделанных свопов. Число m должно удовлетворять условию 0 ≤ m ≤ 4n. Следующие m строк должны содержать по 2 числа: индексы i и j сделанной перестановки двух элементов, индексы считаются с 0. После всех перестановок в нужном порядке массив должен стать пирамидой, то есть для каждого i при 0 ≤ i ≤ n−1 должны выполняться условия:

1. если 2i + 1 ≤ n − 1, то ai < a2i+1
2. если 2i + 2 ≤ n − 1, то ai < a2i+2

* Обратите внимание, что все элементы входного массива различны. Любая последовательность свопов, которая менее 4n и после которой входной массив становится корректной пирамидой, считается верной;
* Ограничение по времени. 3 сек.;
* Ограничение по памяти. 512 мб.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

**Задание 7.**

Напишите программу пирамидальной сортировки на Python для последовательности в убывающем порядке. Проверьте ее, создав несколько рандомных массивов, подходящих под параметры:

* Формат входного файла (input.txt). В первой строке входного файла содержится число n (1 ≤ n ≤ 105) — число элементов в массиве. Во второй строке находятся n различных целых чисел, по модулю не превосходящих 109;
* Формат выходного файла (output.txt). Одна строка выходного файла с отсортированным по невозрастанию массивом. Между любыми двумя числами должен стоять ровно один пробел.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст, табличка

Автоматически созданное описание