МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Н. Э. Баумана

КАФЕДРА ПРОЕКТИРОВАНИЕ И ТЕХНОЛОГИЯ ПРОИЗВОДСТВА ЭЛЕКТРОННОЙ АППАРАТУРЫ

Отчет о выполнении

практического задания

«Изучение метода пирамидальной сортировки графа»

по курсу «Функциональная логика и теория алгоритмов»

Выполнили: студенты гр. ИУ4-21Б

Глухов Степан

Царёв Матвей

Титов Андрей

Юров Всеволод

Лебединцев Фёдор

Проверил: д.т.н., профессор, профессор каф. ИУ-4

Л.А. Зинченко

Цель работы: изучить алгоритмы сортировки, приобрести практические знания в реализации данных алгоритмов, выявить плюсы и минусы. Сравнить данные

Исходные данные:

Язык программирования C.

Массив данных типа integer.

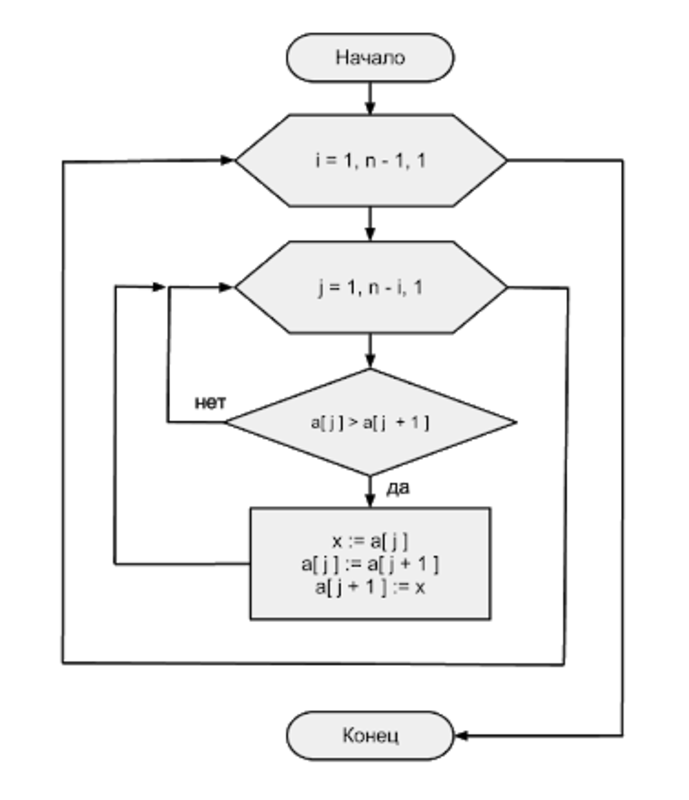
**Обменная сортировка(пузырьковая)**

Сортировка пузырьком (обменная сортировка) **–** простой в реализации и малоэффективный алгоритм сортировки. Метод изучается одним из первых на курсе теории алгоритмов, в то время как на практике используется очень редко.

Упорядоченный массив создается на том же участке памяти, где находится исходная последовательность. Идея метода состоит в том, чтобы попарно сравнивать соседние элементы. Каждый проход начинается с начала последовательности. Сравнивается первый элемент со вторым: если порядок между ними нарушен, то они меняются местами. Затем сравниваются второй с третьим, третий с четвертым и так далее до конца массива; элементы с неправильным порядком в паре меняются местами. В итоге, после первого прохода, максимальный (или минимальный, в зависимости от вида сортировки: по возрастанию/по убыванию) элемент будет находиться на последнем месте в массиве, он как бы «всплывет» наверх. Именно поэтому этот метод называется сортировка пузырьком. На следующем проходе рассматривается последовательность от 1 до N-1, затем от 1 до N-2, и так до конца. После каждого прохода можно делать проверку: выполнялись ли перестановки элементов. Если не выполнялись, то сортировка завершена.

При сортировке пузырьком максимальное число сравнений составляет 0,5N², а среднее число сравнений приблизительно 0,25N².

Схема алгоритма:



**Пирамидальная сортировка**

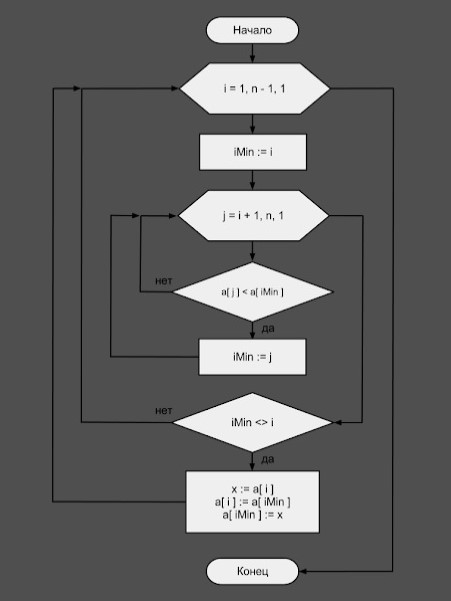
Пирамидальная сортировка (или сортировка кучей, HeapSort) — это метод сортировки сравнением, основанный на такой структуре данных как двоичная куча. Она похожа на сортировку выбором, где мы сначала ищем максимальный элемент и помещаем его в конец. Далее мы повторяем ту же операцию для оставшихся элементов.

Двоичная куча — это законченное двоичное дерево, в котором элементы хранятся в особом порядке: значение в родительском узле больше (или меньше) значений в его двух дочерних узлах. Первый вариант называется max-heap, а второй — min-heap. Куча может быть представлена двоичным деревом или массивом.

Дочерние элементы делятся на левый и правый(в нашем коде на left и right).

Левый элемент имеет индекс 2\*i + 1. Правый элемент иммет индекс 2\*i + 1.

Сортировка пирамидальная похожа на сортировку выбором.Блок схема:



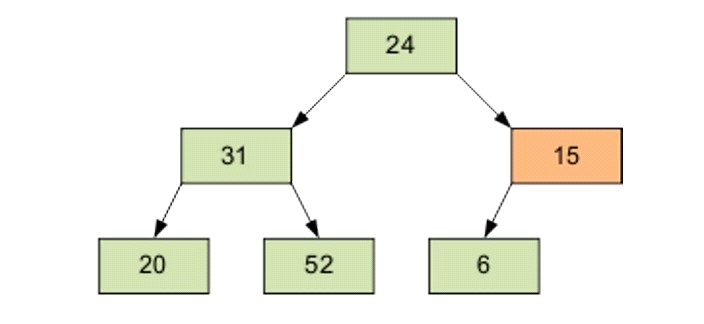
Этап №1.

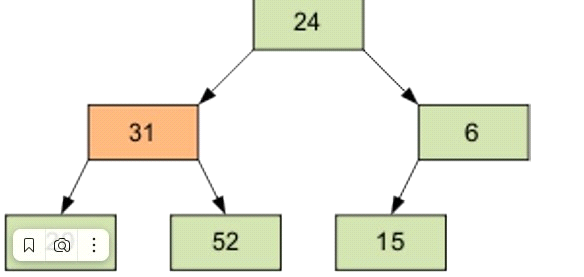
Построение пирамиды. Определяем правую часть дерева, начиная с n/2-1 (нижний уровень дерева). Берем элемент левее этой части массива и просеиваем его сквозь пирамиду по пути, где находятся меньшие его элементы, которые одновременно поднимаются вверх; из двух возможных путей выбираете путь через меньший элемент.

Этап №2.

Сортировка на построенной пирамиде. Берем последний элемент массива в качестве текущего. Меняем верхний (наименьший) элемент массива и текущий местами. Текущий элемент (он теперь верхний) просеиваем сквозь n-1 элементную пирамиду. Затем берем предпоследний элемент и т.д.

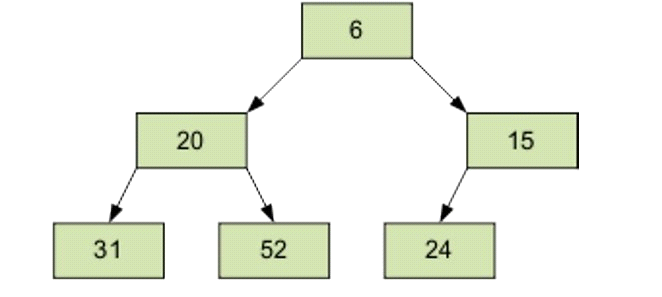
Например: возьмем массив a[n]={24, 31, 15, 20, 52, 6}, тогда первый этап будет выглядить так:





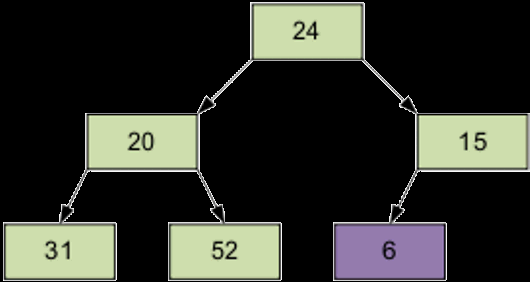
и т.д.

Итог :



Разумеется, полученный массив еще не упорядочен. Однако процедура просеивания является основой для пирамидальной сортировки. В итоге просеивания наименьший элемент оказывается на вершине пирамиды.

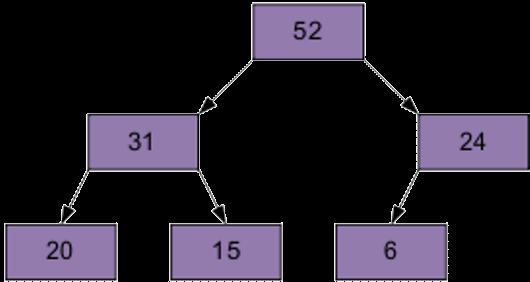
Этап №2.



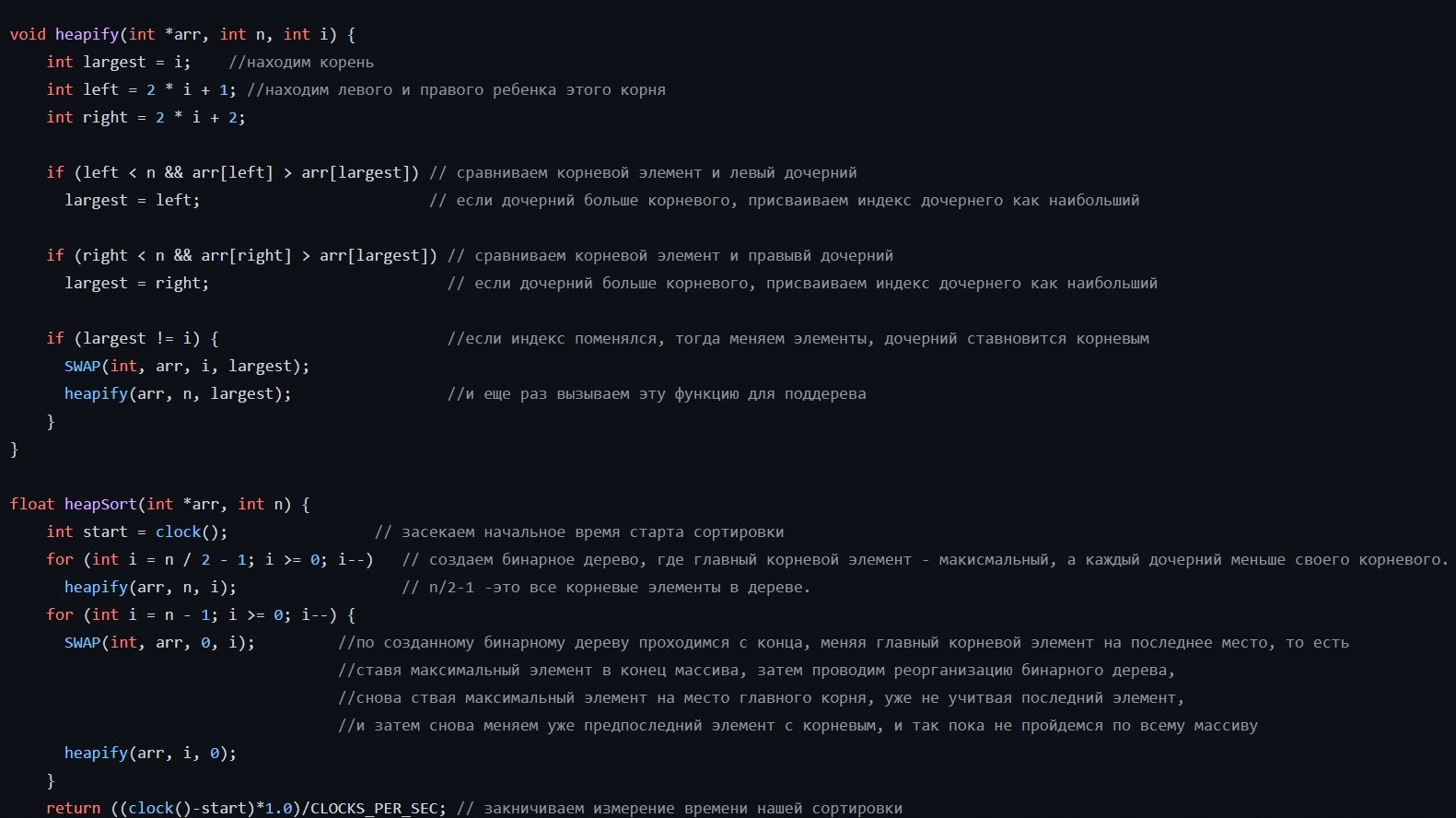
Продолжая данный процесс массив будет отсортирован по убыванию. Такой же принцип присутствует при построении графа.

3. Результаты работы.

Как итог мы получим граф - дерево, в котором вершины идут в порядке убывания.



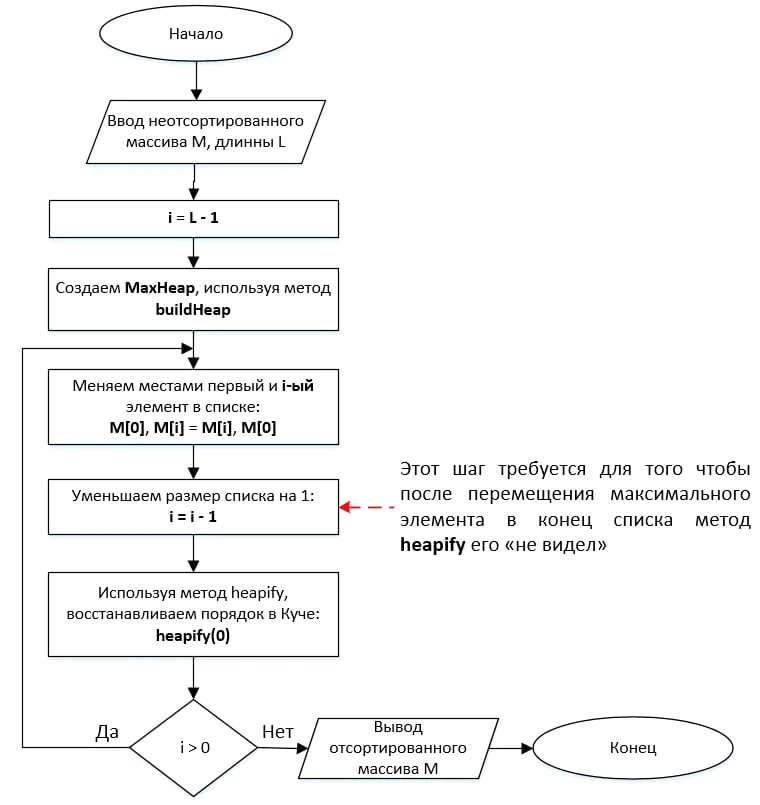
Часть кода ответственная за это преобразование:



Благодаря маркосу SWAP мы производим замену элементов, то есть дочерний элемент может перейти на место корнегого и наоборот.

А функция heapSort применяется для реализации перехода от этапа №1 к этапу №2.

Используя словесную схему для пирамидального метода, можно вкраце объяснить процесс сортировки.



Возьмем частный случай где n=300000. Тогда количество байт занимаемое этим множеством элементов типа int будет 1200000 байт или 1,2 мегабайт(МБ), количество шагов будет 5458380. Данный метод может показаться громоздким, но весьма эффективным для такого количества данных.

4. Вывод: изучив метод пирамидальной сортировки графа(массива), можно заметить, что данный метод предпочтительно использовать только в тех случаях, когда количество элементов заданного массива a[n] очень велико. Для более меньшего количества элементов данный метод рассматривать не стоит. В худшем случае требуется n\*log2(n) шагов, сдвигающих элементы. Среднее число перемещений примерно равно (n/2)\*log2(n).