МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологии

Кафедра Информационных систем и технологий

Специальность 1-40 05 01 Информационные системы и технологии

**РЕФЕРАТ**

на тему:

«Пирамидальная сортировка»

Выполнил студент Iкурса 3 группы специальности ИСиТ Задорожный Н.П

(Ф.И.О)

Руководитель

(учен. степень, звание, должность, подпись, Ф.И.О.)

Минск 2019

Метод пирамидальной сортировки является улучшением традиционных сортировок с помощью дерева. Нотация Big O для пирамидальной сортировки O(n log n).  Количество применяемой служебной памяти не зависит от размера массива (то есть, O(1)). Пирамидальная сортировка не является устойчивой.

Функция для формирования кучи(Пирамиды):  
// Функция формирование кучи

void sift(int \*numbers, int root, int bottom)

{

int maxChild; // индекс максимального потомка

int flagDone = 0; // флаг, потверждающий что куча сформирована

// Пока не дошли до последнего ряда

while ((root \* 2 <= bottom) && (!flagDone))

{

if (root \* 2 == bottom) // если мы в последнем ряду,

maxChild = root \* 2; // запоминаем левый потомок

// иначе запоминаем больший потомок из двух

else if (numbers[root \* 2] > numbers[root \* 2 + 1])

maxChild = root \* 2;

else

maxChild = root \* 2 + 1;

// если элемент вершины меньше максимального потомка

if (numbers[root] < numbers[maxChild])

{

int temporary = numbers[root]; // меняем их местами

numbers[root] = numbers[maxChild];

numbers[maxChild] = temporary;

root = maxChild;

}

else // иначе пирамида готова

flagDone = 1; // пирамида готова

}

}

Функция, которая служит для сортировки самой кучи:

// Функция сортировки на куче

void heapSort(int \*numbers, int ArraySize)

{

// Формируем нижний ряд пирамиды

for (int i = (ArraySize / 2) - 1; i >= 0; i--)

sift(numbers, i, ArraySize - 1);

// Просеиваем через пирамиду остальные элементы

for (int i = ArraySize - 1; i >= 1; i--)

{

int temporary = numbers[0];

numbers[0] = numbers[i];

numbers[i] = temporary;

sift(numbers, 0, i - 1);

}

}

Функция main:

int main()

{

int a[10];

// Заполнение массива случайными числами

for (int i = 0; i < 10; i++)

a[i] = rand() % 20;

// Вывод элементов массива до сортировки

for (int i = 0; i < 10; i++)

cout << a[i] << ' ';

cout << endl;

heapSort(a, 10); // вызов функции сортировки

// Результат

for (int i = 0; i < 10; i++)

cout << a[i] << ' ';

cout << endl;

system("pause");

return 0;

}

Результаты выполнения сортировки:  
