

**Отчет по лабораторной работе №3  
«Сортировки за  $O(n \log n)$ - сравнение  
пирамидальной сортировки и сортировки  
слиянием»**

**Выполнил: студент группы Р3117**

**Плюхин Дмитрий**

**Проверил: Симоненко З. Г.**

**2016 год**

## 1. Задание

Реализовать на каком-либо языке программирования и сравнить между собой алгоритм пирамидальной сортировки и алгоритм сортировки слиянием.

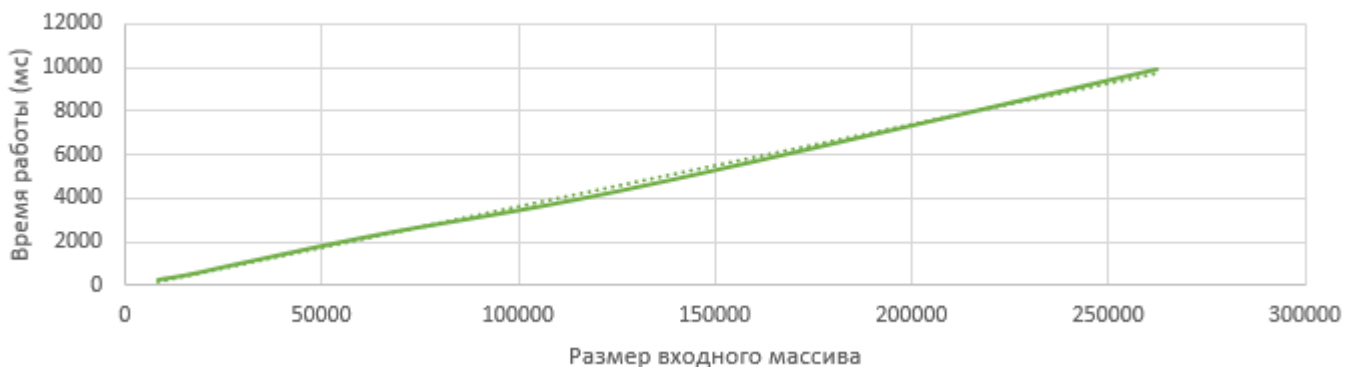
## 2. Выполнение

Для выполнения работы был выбран язык программирования Python по причине того, что алгоритмы сортировок, реализуемых в работе, имеют небольшой объем и не очень сложны с технической точки зрения.

После реализации алгоритмов на языке программирования был проведен их запуск на различных исходных данных, в частности, на массивах разной длины. Была выполнена сортировка массивов с использованием алгоритмов (время сортировки усреднено для каждого массива), построены графики, отражающие зависимость времени работы каждого алгоритма от количества элементов в сортируемом массиве. Каждая сортировка представлена только одним графиком, поскольку время работы сортировок, рассматриваемых в данной работе, на всех перестановках исходного массива в среднем одинаково.

## 3. Результаты

Пирамидальная сортировка

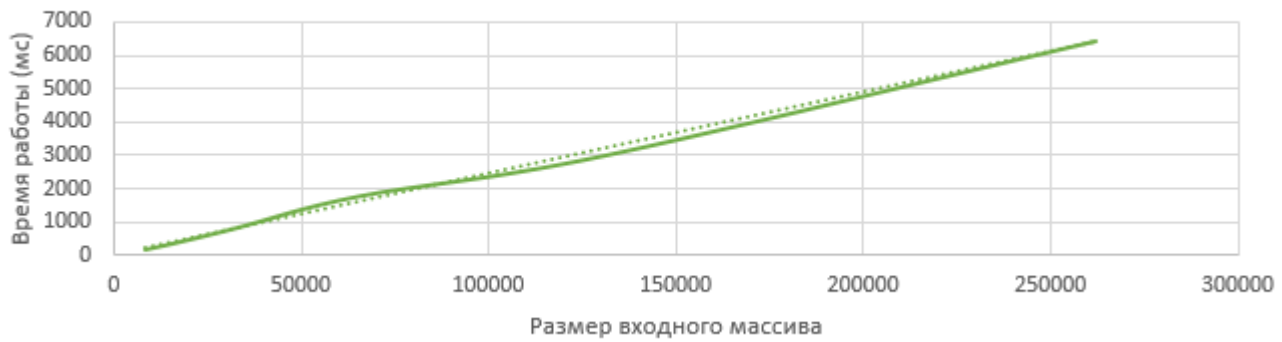


```
import time

def SiftDown(a,i,n):
    while (i < n//2):
        child = i*2 + 1
        if (child < n - 1):
            if (a[child] < a[child + 1]):
                child += 1
        if a[child] < a[i]:
            break
        a[child],a[i] = a[i],a[child]
        i = child

def HeapSort(k):
    for i in reversed(range(0,len(k)//2)):
        SiftDown(k,i,len(k))
    for i in reversed(range(0,len(k))):
        k[i],k[0] = k[0],k[i]
        SiftDown(k,0,i)
```

## Сортировка слиянием

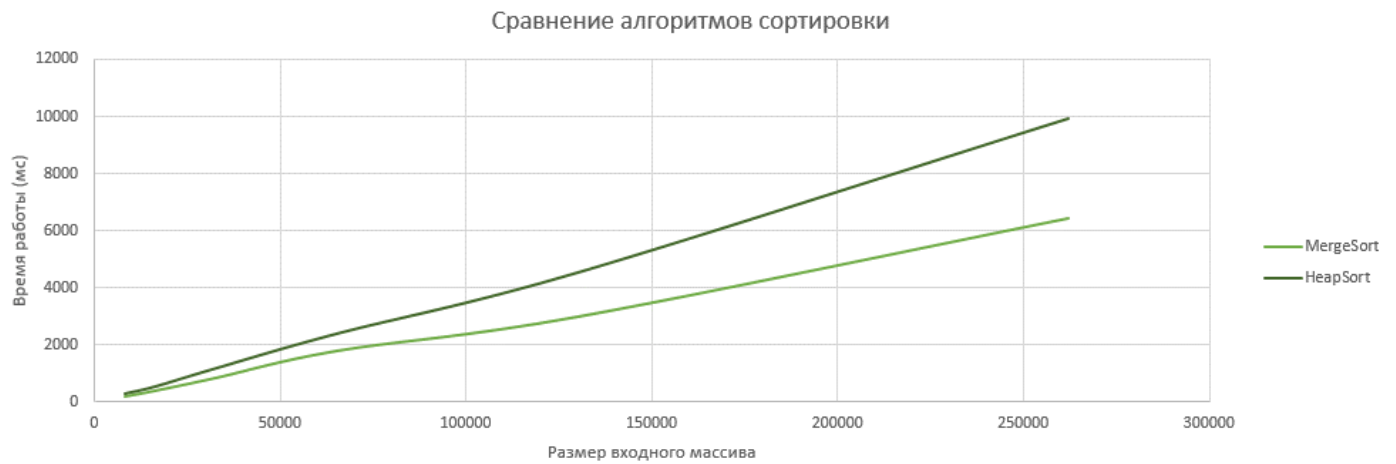


```
import time;
def Merge(a,b):
    n = len(a)
    m = len(b)
    i = 0
    j = 0
    union = []
    for k in range(n+m):
        if i >= n:
            union.append(b[j])
            j+=1
        elif j>= m:
            union.append(a[i])
            i+=1
        elif (a[i]<b[j]):
            union.append(a[i])
            i+=1
        else:
            union.append(b[j])
            j+=1
    return union

def MergeSort(a):
    if len(a) <= 1:
        return a
    return Merge (MergeSort (a[:len(a)//2]),MergeSort (a[len(a)//2:]))
```

## 4. Анализ

Время работы (мс)		Размер массива
Пирамидальная сортировка	Сортировка слиянием	
265	188	8192
515	391	16384
1156	843	32768
2375	1796	65536
4546	3015	131072
9906	6453	262144



Исходя из данных, полученных экспериментальным путем, можно сделать вывод о том, что сортировка слиянием оказывается несколько эффективнее, чем пирамидальная сортировка. Однако, у сортировки слиянием есть недостаток, связанный с тем, что алгоритм использует большое количество указателей при разбиении массива на части и рекурсивном запуске. Сортировка кучей в этом плане эффективнее сортировки слиянием, так как требует память только под исходный массив.