Санкт-Петербургский политехнический университет Петра Великого
Институт машиностроения, материалов и транспорта

Курсовая работа

Дисциплина: Объектно-ориентированное программирование

Тема: Сортировка слиянием

Выполнил студент группы 3331506/80401: Резец Ю. А.

Преподаватель: Ананьевский М. С.

« » 2021 г.

Санкт-Петербург

2021

Сортировка слиянием – алгоритм сортировки, который упорядочивает списки (или другие структуры данных, доступ к элементам, которых можно получать только последовательно) в определённом порядке.

Суть алгоритма состоит в следующем: допустим у нас есть 2 отсортированных по возрастанию массива. Чтобы получить из них общий отсортированный массив, необходимо сравнивать первые элементы массивов и записывать наименьший из них третий массив. Идея сортировки слиянием состоит в том, чтобы рекурсивно разбивать исходный массив, пока не останутся массивы длиной один, поскольку единичный массив является отсортированным, мы можем применить к нему метод слияния, описанный выше и в итоге получить отсортированный массив.

Скорость работы алгоритма:

Лучшее время: $O(n\ log(n))$

Худшее время: $O(n\ log(n))$

Среднее время: $O(n\ log(n))$

На рисунке 1 представлен график зависимости времени сортировки массива, от количества элементов в этом массиве.

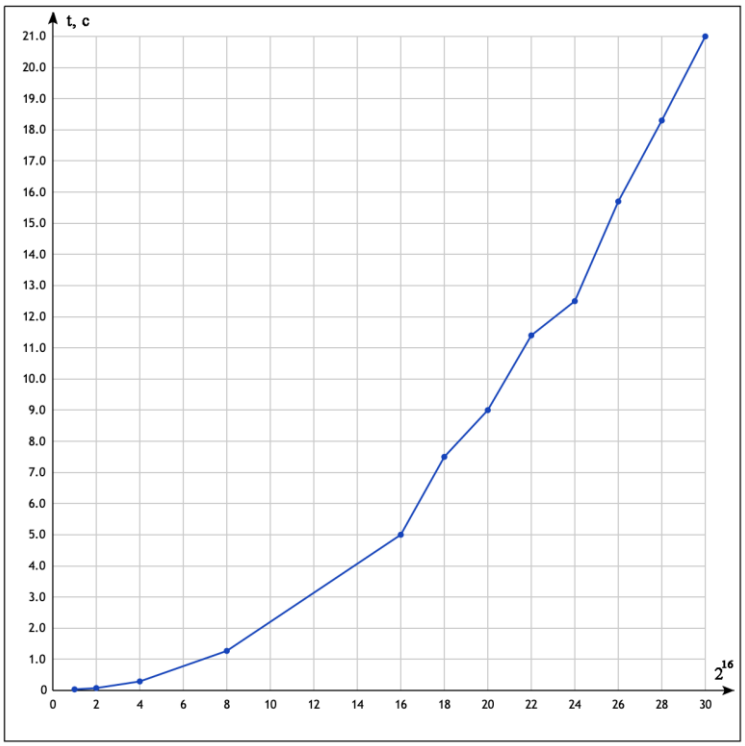

Рисунок 1

Список литературы

1. Левитин, А. В. Алгоритмы. Введение в разработку и анализ / М. Вильямс, 2006.

2. Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К. Алгоритмы: построение и анализ / М. Вильямс, 2005.

3. Бхаргава А., Грокаем Алгоритмы / Питер, 2017.

Приложение

```
#include <iostream>

void merge_array(int *array, int first_index, int last_index)
{
   int *temp_array = new int[last_index + 1];
   int middle = (first_index + last_index)/2;
   int array1_index = first_index;
   int array2_index = middle + 1;

   for (int i = first_index; i <= last_index; i++)
   {
      if ((array1_index <= middle) && ((array2_index > last_index) || (array[array1_index]
< array[array2_index])))
      {
         temp_array[i] = array[array1_index];
         array1_index++;
      }
      else
      {
         temp_array[i] = array[array2_index];
         array2_index++;
      }
   }
   for (int i = first_index; i <= last_index; i++)
      array[i] = temp_array[i];
   delete []temp_array;
}

void merge_sort_array(int *array, int first_index, int last_index)
{
      if (first_index < last_index)
      {
         merge_sort_array(array, first_index, (first_index + last_index)/2);
         merge_sort_array(array, (first_index + last_index)/2 + 1, last_index);
         merge_array(array, first_index, last_index);
      }
}

void merge_sort(int *array, int size)
{
   merge_sort_array(array, 0, size - 1);
}

int main()
{
   return 0;
}
```