Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7**

Дополнительные задания

Вариант № 7

Преподаватель \_\_\_\_\_\_\_\_\_\_ И. В. Матковский

подпись, дата

Студент КИ20-08Б 032049025 \_\_\_\_\_\_\_\_\_\_ А. Н. Головань

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2021

**1 Цель работы**

Научиться использовать алгоритмы внешней сортировки.

**2 Порядок выполнения работы**

Используя один из описанных в лекции алгоритмов внешней сортировки, отсортировать массив. Исходный массив целых чисел считывается из входного файла. Количество элементов в массиве может превышать 1012. Для демонстрации программы необходимо её доработать так, чтоб она могла вернуть элемент уже отсортированного массива по его индексу.

**3 Задание**

Реализовать алгоритм внешней сортировки “Прямое слияние”

**4 Код программы**

P5.py

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  147  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175 | from test import \*  def vertical(fileName):  with open(fileName,"r") as file:  text = file.read()  text = text.replace(" ","\n")  with open(fileName,"w") as file:  file.write(text)  def fileAppend(fileName,value):  with open(fileName,"r") as file:  text = file.read()  if text:  value = " " + value  with open(fileName,"a") as file:  file.write(value)  def fileClear(fileName):  with open(fileName,"w") as file:  file.write("")  # with open("A.txt","w") as file:  # file.write("8 2 23 27 5 3 65 68 44 42 33 40 1 0 6 10 2")  # fileClear("B.txt")  # fileClear("C.txt")  def merge(left, right):  # Если первый массив пуст, то ничего не нужно  # для объединения, и вы можете вернуть второй массив в качестве результата  if len(left) == 0:  right = str(right).replace("[","")  right = str(right).replace("]","")  return "[" + str(right).replace(" ","") + "]"  # Если второй массив пуст, то ничего не нужно  # для объединения, и вы можете вернуть первый массив как результат  if len(right) == 0:  left = str(left).replace("[","")  left = str(left).replace("]","")  return "[" + str(left).replace(" ","") + "]"  result = []  index\_left = index\_right = 0  text = ""  # Теперь перебираем оба массива, пока все элементы  # превратить его в результирующий массив  while len(result) < len(left) + len(right):  # Элементы необходимо отсортировать, чтобы добавить их в  # результирующий массив, поэтому вам нужно решить, получать ли  # следующий элемент из первого или второго массива  if left[index\_left] <= right[index\_right]:  text = text + str(left[index\_left]) + ", "  # result.append(left[index\_left])  index\_left += 1  else:  text = text + str(right[index\_right]) + ", "  # result.append(right[index\_right])  index\_right += 1  # Если вы дойдете до конца любого массива, вы можете  # добавляем оставшиеся элементы из другого массива в  # результат и разорвать цикл  if index\_right == len(right):  for i in left[index\_left:]:  text = text + str(i) + ", "  # result += left[index\_left:]  break  if index\_left == len(left):  for i in right[index\_right:]:  text = text + str(i) + ", "  # result += right[index\_right:]  break  text = "[" + text.replace(" ","")[:-1] + "]"  return text  def transResult():  with open("A.txt","r") as file:  text = file.read()    text = text.replace(","," ")  text = text.replace("[","")  text = text.replace("]","")  with open("A.txt","w") as file:  file.write(text)  def scatterValues():  vertical("A.txt")  with open("A.txt","r") as file:  line = "123"  count = 0  while line:  line = file.readline().replace("\n","")  if not line:  continue  count += 1  if count%2==1:  fileAppend("B.txt",line)  else:  fileAppend("C.txt",line)  fileClear("A.txt")  def getCountEl(fileName):  vertical(fileName)  with open(fileName,"r") as file:  line = "123"  count = 0  while line:  line = file.readline().replace("\n","")  if not line:  continue  count += 1    return count    def sorterBC():  fileB = open("B.txt", "r")  fileC = open("C.txt", "r")  vertical("B.txt")  vertical("C.txt")  while True:  flag = False  multy = False  lineB = fileB.readline().replace("\n","")  lineC = fileC.readline().replace("\n","")  if "[" in lineB:  multy = True    if not lineB or not lineC:  flag = True  if not lineB and not lineC:  break  if multy==False:  # print("{} - {}".format(lineB,lineC))  if lineB > lineC:  text = "[" + lineC + "," + lineB + "]"  else:  text = "[" + lineB + "," + lineC + "]"    if flag:  text = text.replace(",","")    fileAppend("A.txt",text)    else:  it = 0  lineB = eval(lineB)  if lineC:  lineC = eval(lineC)  else:  lineC = []    if isinstance(lineB,int): #если множество заданий  lineB = [lineB]    if isinstance(lineC,int): #если множество заданий  lineC = [lineC]  text = "["  print("lineB -->{}".format(lineB))  print("lineC -->{}".format(lineC))  text = merge(sorted(lineB),sorted(lineC))  print(text)  print(" ")  # sorting = sorting.replace("[","(")  # sorting = sorting.replace("]",")")  fileAppend("A.txt",text)    fileClear("B.txt")  fileClear("C.txt")  fileB.close()  fileC.close()    flag = False  while True:  scatterValues()  nB = getCountEl("B.txt")  if nB == 1:  flag = True  sorterBC()  if flag == True:  transResult()  break  with open("A.txt","r") as file:  save = file.read()  number = 3  vertical("A.txt")  line="123"  with open("A.txt","r") as file:  number+= 1  while line and number>0:  number-=1  line = file.readline()  with open("A.txt","w") as file:  file.write(save)  print(line)  print("file was sorted") |

**5 Теоретические оценки сложности алгоритмов**

В программе используются алгоритм внешней сортировки “Прямое слияние”. Алгоритм “Прямое слияние” и его сложность составляет O(n\*log( n ))

**6 Экспериментальные оценки сложности алгоритмов**

Таблица 1 - зависимость времени от объема входных данных

|  |  |  |  |
| --- | --- | --- | --- |
| Количество элементов | 20 | 50 | 100 |
| Прямое слияние | 0.0269с | 0.056с | 0.085с |