Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 1 з дисципліни
«Проектування алгоритмів»

«Проектування алгоритмів зовнішнього сортування»

Варіант 4

Виконав студент ІП-15, Буяло Дмитро Олександрович
(шифр, прізвище, ім'я, по батькові)

Перевірив Соколовський Владислав Володимирович
(прізвище, ім'я, по батькові)

Київ 2022

ЗМІСТ

1. МЕТА ЛАБОРАТОРНОЇ РОБОТИ 6
2. ЗАВДАННЯ
3. ВИКОНАННЯ
   3.1.ПСЕВДОКОД АЛГОРИТМУ
   3.2.ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ
      3.2.1. Вихідний код

ВИСНОВОК

1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Мета роботи – вивчити основні алгоритми зовнішнього сортування та способи їх модифікації, оцінити поріг їх ефективності.

Індивідуальне завдання

Варіант 4

2 ЗАВДАННЯ

Для алгоритму багатофазного сортування розробити та записати алгоритм зовнішнього сортування за допомогою псевдокоду (чи іншого способу за вибором).

Виконати програмну реалізацію алгоритму на будь-якій мові програмування та відсортувати випадковим чином згенерований масив цілих чисел, що зберігається у файлі (розмір файлу має бути не менше 10 Мб, можна значно більше).

Здійснити модифікацію програми і відсортувати випадковим чином згенерований масив цілих чисел, що зберігається у файлі розміром не менше ніж 32Гб. Досягти швидкості сортування з розрахунку 1Гб на 3хв або менше.

Рекомендується попередньо впорядкувати серії елементів довжиною, що займає не менше 100Мб або використати інші підходи для пришвидшення процесу сортування.

Зробити узагальнений висновок з лабораторної роботи, у якому порівняти базову та модифіковану програми. У висновку деталізувати, які саме модифікації було виконано і який ефект вони дали.

3 ВИКОНАННЯ

3.1 Псевдокод алгоритму

```
POLYPHASE_STEP(s, N)
    (l, F, d) ← HORIZONT_DISTR_M(s, N)
    while l > 0 do
        while d[N – 1] ≠ < > И F[N – 1] ≠ 0 do
            k1 ← k2 ← 0
            for i ← 1 to N – 1 do
                if F[i] > 0 then  k1 ← k1 + 1, m1[k1] ← i
                else              k2 ← k2 + 1, m2[k2] ← i
            if k1 = N – 1 then
                F[N] ← F[N] + 1
            else
                MERGERUN_N(d[N], k2, d[m2[1]], …, d[m2[k2]])
            for k ← 1 to k1 do
                F[m1[k]] ← F[m1[k]] – 1
        l ← l – 1
        (d[1], d[2], d[3],…, d[N]) ← (d[N], d[1], d[2], …, d[N – 1])
    return d[1]
```



```
HORIZONT_DISTR_M(s, N)
    for i = 1 to N – 1 do
        A[i] ← F[i] ← 1, d[i] ← < >
    A[N] ← F[N] ← 1, d[N] ← < >, l ← j ← 1
    while a ≠ < > do
        last[j] ← COPYRUN(a, d[j]),
        F[j] ← F[j] – 1
        if  F[j] < F[j + 1] then  j ← j + 1
        else if F[j] ≠ 0 then j ← 1
        else
            l ← l + 1, x ← A[1]
            for i = 1 to N – 1 do
                F[i] ← x + A[i + 1] – A[i]
                A[i] ← x + A[i + 1]
            j ← 1
        CHECKMERGE(a, d[j], last[j])
    return (l, F, d)
```

```
COPYRUN(a, b)
    cur ← first(a), a ← rest(a)
    if a ≠ < > then
        next ← first(a)
    else
        next ← cur – 1
    while a ≠ < > И next ≥ cur do
        b ← b & cur
        cur ← next
        a ← rest(a)
        if a ≠ < > then
            next ← first(a)
    b ← b & cur
    return cur
```

```
CHECKMERGE(a, d, last)
    if a ≠ < > then
        if first(a) > last then
            COPYRUN(a, d)
```

```
MERGERUN_N(d, N, s_1, …, s_N )
    k ← 0
    for i = 1 to N do
        if s_i ≠ < > then   c_i ← first(s_i), k ← k + 1, m_k ← i
    k' ← (k > 0)
    while k > 0 do
        j' ← 1, i' ← m_1, c' ← c_i'
        for j = 2 to k do
            i ← m_j
            if  c_i < c' then j' ← j, i' ← m_j, c' ← c_i'
        d ← d & c',   s_i' ← rest(s_i')
        if s_i' = < > ИЛИ c_i' ≥ first(s_i')  then
            m_k ↔ m_j' , k ← k – 1
        else c_i' ← first(s_i')
    return (k', d)
```

m – карта непустых серий
m_1 – первая непустая серия
m_k – последняя непустая серия

для удаления последовательности с пустой серией достаточно поменять ее местами с последней непустой серией и уменьшить количество k непустых серий

3.2 Програмна реалізація алгоритму

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.util.Random;
import java.util.Scanner;

import static java.lang.Integer.MAX_VALUE;

public class Main {
   public static void main(String[] args) throws IOException {
      System.out.println("Hello world!");System.out.println("Hello world!");
      Scanner scan = new Scanner(System.in);
      System.out.print("Enter the name of file: ");
      String name = scan.nextLine() + ".txt";
      System.out.print("In which number system to create a file? (Kb/Mb/Gb): ");
      String system = scan.nextLine().toUpperCase();
      System.out.print("Enter file size in " + system + ": ");
      long size = scan.nextLong();
      long time = System.currentTimeMillis();
      System.out.print("A file with the name \"" + name + "\" and size of " + size + system +
            " was successfully generated in ");
      System.out.println((double) (System.currentTimeMillis() - time) + "ms!");
      System.out.print("\nHow many files to use for sorting? N = ");
      int n = scan.nextInt();
      time = System.currentTimeMillis();
      System.out.print("Polyphase sorting was done in");
      System.out.println((double) (System.currentTimeMillis() - time) + "ms!");
      System.out.println("The result in T"+i+1+" file");
      polyPhaseSort();
   }
   public static void select() {
//       int i, z;
      if(d[j]<d[j+1]) j=j+1;
      else {
         if(d[j]==0) {
            level = level+1;
            z=a[i];
            for (i=1;i<=n;i++) {//!!!
               d[i]=z+a[i+1]-a[i];
               a[i]=z+a[i+1];
            }
         }
```

```
        j=0;//!!!
    }
    d[j] = d[j]-1;
}
public static void copyrun(File f0, File file) {
    do {
        copy();
    } while (true);//!?
}
public static void openRandomSeq(String system, long size) throws IOException {
    File file = new File("name.txt");
    file.createNewFile();
    FileWriter in = new FileWriter(file);
    Random rand = new Random();
    StringBuilder s = new StringBuilder();
    long count =0;
    switch (system) {
        case "GB":
            size *= 1024 * 1024 * 1024;
            break;
        case "MB":
            size *= 1024 * 1024;
            break;
        case "KB":
            size *= 1024;
            break;
    }
    while (file.length() < size) {
        for (int i = 0; i < 10240; i++) {
            s.append(rand.nextInt(MAX_VALUE));
            s.append('\n');
        }
        in.write(s.toString());
        s.setLength(0);
        count++;
    }
    in.close();
}
public static void startWrite() {

}
public static void copy() {

}
private static void openSeq(File file, String name) {

}
public static final int n=6;
public static int i,j,z,level;
public static int[] a = new int[n];
```

```
   public static int[] d = new int[n];
   public static void polyPhaseSort() throws IOException {
//       final int n =6;
      int mx,tn;
      int k,dn;
      int x,min;
      int[] t = new int[n];
      int[] ta = new int[n];
      File f0;
      File[] f = new File[n];
      String[] fn = new String[n];
      int eof = 5;
      int c=0;

      FileWriter[] in = new FileWriter[n];

//       openRandomSeq("gb",3);
      f0 = new File("name.txt");
      //listSeq();
//          openSeq(f[i],fn[i]);
      for (i=0;i<n;i++) {
         fn[i]="T"+(i+1)+".txt";
         f[i] = new File(fn[i]);
         f[i].createNewFile();
      }
      for (i=0;i<n-1;i++) {
         a[i]=1;d[i]=1;
         //startWrite(f[i]);//!?
         in[i]= new FileWriter(f[i]);
      }
      level=1;j=0;//!!!
      a[n-1]=0;d[n-1]=0;//!!!
//       startRead(f0);
      BufferedReader reader = Files.newBufferedReader(f0.toPath(), StandardCharsets.UTF_8);
      boolean eor;
      String firstX, firstZ;
      String[] firstY=new String[n];
      firstX = reader.readLine();
      do {
         select();
//          copyrun(f0,f[j]);
         do {
//             copy();
            firstY[j]=firstX;
            in[j].write(firstX+'\n');
            firstX = reader.readLine();
            eor=false;
            if(Integer.parseInt(firstX)<Integer.parseInt(firstY[j])) eor=true;
            c++;
         } while (!eor);//!?
```

```
    } while (c!=eof && (j!=n-2));//!?
    j=0;
    while (c!=eof) {//!?
      select();
      if(Integer.parseInt(firstY[j]) <= Integer.parseInt(firstX)) {
//        copyrun(f0,f[j]);
        do {
//          copy();
          firstY[j]=firstX;
          in[j].write(firstX+'\n');
          firstX = reader.readLine();
          eor=false;
          if(Integer.parseInt(firstX)<Integer.parseInt(firstY[j])) eor=true;
          c++;
        } while (!eor);//!?
        if(c!=eof) {//!?
          d[j]=d[j]+1;
        }
//        copyrun(f0,f[j]);
        do {
//          copy();
          firstY[j]=firstX;
          in[j].write(firstX+'\n');
          firstX = reader.readLine();
          eor=false;
          if(Integer.parseInt(firstX)<Integer.parseInt(firstY[j])) eor=true;
        } while (!eor);//!?
      }
//      copyrun(f0,f[j]);
      do {
//          copy();
        firstY[j]=firstX;
        in[j].write(firstX+'\n');
        firstX = reader.readLine();
        eor=false;
        if(Integer.parseInt(firstX)<Integer.parseInt(firstY[j])) eor=true;
      } while (!eor);//!?
    }
    BufferedReader[] readers = new BufferedReader[n];
    for(i=0;i<n-1;i++) {//!!!
      t[i] = i;
//      startRead();
      readers[i] = Files.newBufferedReader(f[i].toPath(), StandardCharsets.UTF_8);
    }
    t[n-1] = n-1;
    do {
      z=a[n-2]; d[n-1]=0;
//      startWrite(f[t[n-1]],t[n-1]);!!!!!!!!!!!!
      do {
        k = 0;
```

```
            for (i = 0; i < n - 1; i++) {//!!!
                if (d[i] > 0) {
                    d[i] = d[i] - 1;
                } else {
                    ta[k] = t[i]; k = k + 1;
                }
            }
            if (k == 0) {
                d[n-1] = d[n-1] + 1;
            }
            else {
                do {
                    i = 0; mx = 0; min = Integer.parseInt(readers[ta[0]].readLine()); //f[ta[0]].first;
                    while (i < k) {
                        i = i + 1; x = Integer.parseInt(readers[ta[i]].readLine());//????
                        if (x < min) {
                            min = x; mx = i;
                        }
                    }
//                  copy(f[ta[mx]],f[t[n-1]]);
                    firstX=readers[ta[mx]].readLine();
                    firstZ=firstX;
                    in[t[n-1]].write(firstX);
                    eor=false;
                    if(Integer.parseInt(firstX)<Integer.parseInt(firstZ)) eor=true;

                    if (eor) {
                        for (int tx = mx; tx < k; tx++) {
                            ta[tx] = ta[tx + 1];
                        }
                        k = k - 1;
                    }
                } while (k != 0);
            }
            z = z - 1;
        }while (z!=0);
//          startRead(f[t[n-1]],t[n-1]);
        tn=t[n-1];dn=d[n-1];z=a[n-2];
        for (i=n-1;i>0;i--) {
            t[i]=t[i-1];d[i]=d[i-1];a[i]=a[i-1]-z;
        }
        t[0]=tn;d[0]=dn;a[0]=z;
//          startWrite(f[t[1]], t[n]);
        in[t[1]].write(t[n-1]);
        level=level-1;
    } while (level!=0);
    for (i=0;i<n;i++) {
//          closeSeq(f[i]);
        in[i].close();
        readers[i].close();
```

```
        }
//        closeSeq(f0);
        reader.close();
        }



    public static void straightMerge(int n, int[] a) {//как в учебнике!
        int i,j, k, l,t;
        int h,m, p ,q,r;
        boolean up;

        up = true;
        p=1;
        do{
            h=1; m=n;
            if(up) {i=0; j=n-1; k=n; l=2*n-1;}
            else   {k=0; l=n-1; i=n; j=2*n-1;}
            do{
                if(m>=p) {q=p;}
                else {q=m;}
                m=m-q;
                if(m>=p) {r=p;}
                else {r=m;}
                m=m-r;
                while ((q!=0) && (r!=0)) {
                    if(a[i]<a[j]){a[k]=a[i];k=k+h;i=i+1;q=q-1;}
                    else         {a[k]=a[j];k=k+h;j=j-1;r=r-1;}
                }
                while (r>0)       {a[k]=a[j];k=k+h;j=j-1;r=r-1;}
                while (q>0)       {a[k]=a[i];k=k+h;i=i+1;q=q-1;}
                h=-h; t=k; k=l; l=t;
            } while (m!=0);
            up=!up;p=2*p;
        } while (p<=n);
        if(!up) {
            for(i=0;i<n;i++) {
                a[i]=a[i+n];
            }
        }
    }
}
```

3.2.1 Вихідний код

```
Hello world!
Enter the name of file: basicAlg
In which number system to create a file? (Kb/Mb/Gb): mb
Enter file size in MB: 10
A file with the name "basicAlg.txt" and size of 10MB was successfully generated in 79.0ms!
How many files to use for sorting? N = 6
Polyphase sorting was done in 3815.0ms!
The result in T1 file
```