提出日　令和2年07月09日

アルゴリズムとデータ構造　第6回課題レポート

学籍番号（A19117）
氏名（永尾優磨）

【課題】

1. 今回のソート方法2種類と、それ以外のもう1種類のソート方法について調査し、説明せよ。
2. バブルソートについて以下の作業を行い、ソースリストと実行結果をレポートにまとめよ。
   - バブルソートクラスとそれを実行するクラスを作成せよ。
   - 適当なデータを放り込み（10個程度）、バブルソートが行えるかどうか確認せよ。

3. クイックソートについても以下の作業を行い、ソースリストと実行結果をレポートにまとめよ。
   - クイックソートクラスとそれを実行するクラスを作成せよ。
   - 適当なデータを放り込み（10個程度）、クイックソートが行えるかどうか確認せよ。

【作成したプログラム】

```
public class Bubble {

    private int[] a;
    private int nElems;

    Bubble(int max) {
        a = new int[max];
        nElems = 0;
    }

    public void insert(int value) {
        a[nElems++] = value;
    }

    public void display() {
        int i;
        for (i = 0; i < nElems; i++) {
            System.out.println(a[i]);

        }
    }

    public void bubbleSort() {
```

```
        int out;
        int in;
        display();
        System.out.println("--------------");
        for (out = nElems - 1; out > 0; out--) {
            for (in = 0; in < out; in++) {
                if (a[in] > a[in + 1]) {
                    int tmp;
                    tmp = a[in];
                    a[in] = a[in + 1];
                    a[in + 1] = tmp;
                }
            }
            display();
            System.out.println("--------------");

        }
    }

}
```

プログラム1.1　Bubble

```
import java.util.Random;

public class BubbleApp {
    public static void main(String[] args) {
         int n = 5;
        Bubble app = new Bubble(n);
        Random rnd = new Random();
        for (int i = 0; i < n - 1; i++) {
            app.insert(rnd.nextInt());
        }
        app.bubbleSort();
    }

}
```

プログラム1.2　BubbleApp

```
public class Quick {
    private int[] a;
    private int nElems;

    Quick(int max) {
        a = new int[max];
        nElems = 0;
    }
```

```
public void insert(int value) {
    a[nElems++] = value;
}

public void display(int l, int r) {
    int i;
    for (i = 0; i < nElems; i++) {
        System.out.print(a[i] + ",");
    }
    System.out.print("l=" + l + ",r=" + r + ",");

}

public void quickSort(int left, int right) {
    int pivot;
    int i;
    int j;
    System.out.println(left + "から" + right + "までソート");
    pivot = a[(left + right) / 2];

    display(left, right);
    System.out.println("pivot = " + pivot);
    i = left;
    j = right;
    for (;;) {
        while (a[i] < pivot) {
            i++;
        }
        while (pivot < a[j]) {
            j--;
        }
        if (i >= j) {
            break;
        }
        int tmp = a[i];
        a[i] = a[j];
        a[j] = tmp;

        i++;
        j--;
    }
    if (left < i - 1) {
        quickSort(left, i - 1);
    }
    if (j + 1 < right) {
        quickSort(j + 1, right);
    }
```

```
    }

}
```

プログラム2.1　Quick

```
import java.util.Random;

public class QuickApp {
    public static void main(String[] args) {
        int n = 15;
        Quick app = new Quick(n);
        Random rnd = new Random();

        for (int i = 0; i < n; i++) {
            app.insert(rnd.nextInt());
        }

        app.display(0, n);

        app.quickSort(0, n - 1);
        System.out.println("---------");
        app.display(0, n - 1);
    }
}
```

プログラム2.2　QuickApp

【プログラムの解説】

Bubbleのdisplayの関数は、授業でのスライドから内容を変更した。シンプルに、1行ずつ文を出力する形に変更し、見やすくするために、プリント文の間に---------と出力する文章を付け加えた。

QuickAppは、Randomなどを用いて、データの生成を容易に出来るようにした。

【結果】

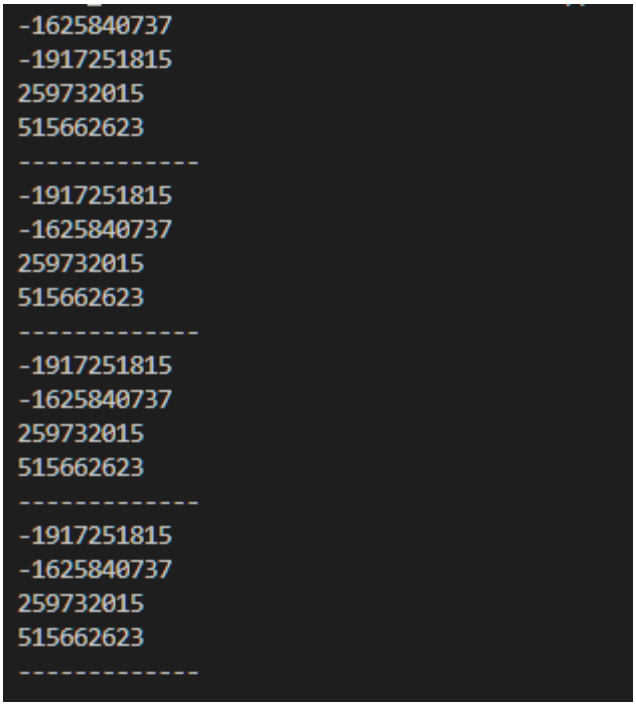

図1　BubbleAppの出力結果

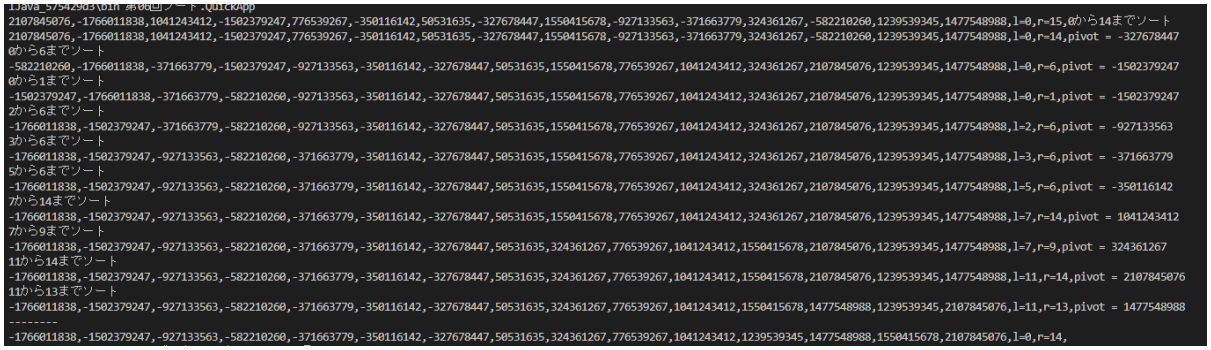

図2　QuickAppの出力結果

【考察】

数字を並べ替える時に真っ先に思いつくのが、Bubbleソートのやり方だったので、今回のQuickソートのやり方を知ることが出来て良かったと思う。裏で、Pythonを用いて、選択ソートやバブルソート、ヒープソート、クイックソートなどを動かしてみたが、クイックソートが一番速いアルゴリズムだった。これからプログラムを書く時には、ライブラリを使うので、どのようなアルゴリズムが動いている事を考えることは無いと思う。今回得た再帰を使ったソートの考え方をすぐにでも使えるようにして、これからのプログラムを書いていきたいと思う。

【参考文献】

1. 「アルゴリズムとデータ解析の授業スライド、線形探索について」
2. 「No enclosing instance of type Hoge is accessible.」〈https://qiita.com/watanabk/items/738988fac29e1e1d8d88〉
3. 増井敏克(2020)「Pythonで始めるアルゴリズム入門」翔泳社
4. 「参考文献の書き方」〈http://www7a.biglobe.ne.jp/nifongo/ron/ron_04.html〉