

提出日 令和2年07月02日

アルゴリズムとデータ構造 第5回課題レポート

学籍番号 (A19117)

氏名 (永尾優磨)

【課題】

1. 今回取り上げた探索手法と、その他の探索手法についても述べよ。
2. 線形探索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。
 - 線形探索用配列を作成せよ。
 - 探索配列に値を設定せよ。
 - 適当な値を探索して結果を表示させよ。
3. 二分探索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。
 - 二分探索用配列を作成せよ。
 - 同様に探索配列に値を設定せよ。
 - 適当な値を探索して結果を表示させよ。

【作成したプログラム】

```
public class SearchArray {
    private int[] array;
    private int size;

    SearchArray(int s) {
        array = new int[s];
        size = s;
    }

    public void setData(int index, int data) {
        array[index] = data;
    }

    public int getData(int index) {
        return array[index];
    }

    public int getSize() {
        return size;
    }

    public int linerSearch(int data) {
        int i;
        for (i = 0; i < size; i++) {
            if (array[i] == data) {
                return i;
            }
        }
    }
}
```

```

        return -1;
    }

    public int binarySerach(int data) {
        int start = 0;
        int end = size - 1;
        int location = -1;
        int middle;

        while (start <= end && location == -1) {
            middle = Math.round((start + end) / 2);
            System.out.println("開始位置 = " + start + ", 中央位置 = " + middle + "終了位置 = " + end);

            if (array[middle] == data) {
                System.out.println("見つかりました");
                location = middle;
            } else if (array[middle] > data) {
                System.out.println("中央より左側にありそうです");
                end = middle - 1;
            } else {
                System.out.println("中央より右側にありそうです");
                start = middle + 1;
            }
        }
        return location;
    }
}

```

プログラム1 配列作成のプログラム

```

public class SearchTest {
    public static void main(String[] args) {
        int arraydata = 5;
        SearchArray array1 = new SearchArray(arraydata);
        int i;
        int j = 1;
        for (i = 0; i < arraydata; i++) {
            array1.setData(i, j);
            j++;
        }
        System.out.println(array1.linerSearch(4));

        int arraydata1 = 50;
        SearchArray array2 = new SearchArray(arraydata1);
        i = 0;
    }
}

```

```

j = 1;
for (i = 0; i < arraydata1; i++) {
    array2.setData(i, j);
    j++;
}

System.out.println(array2.binarySearch(20));
}
}

```

プログラム2 出力のプログラム

【プログラムの解説】

1. 「今回取り上げた探索手法と、その他の探索手法」

1.1. 今回取り上げた探索手法について

線形探索

線形探索とは、最も単純なアルゴリズムである。流れとしては、以下の通りである。

1. 配列などに格納されたデータの中から、まず戦闘の要素を探しているデータと比較する。
 2. 一致しなければ2番目の要素と比較する。
 3. 1, 2を繰り返し、途中でデータを発見したら、その位置を返す
- 最も単純なアルゴリズムと称される通り、行っていることは、1, 2, 3の通りシンプルである。

利点としては、先頭から比較するため、事前にデータ列の整列などを行う必要が無いことが挙げられる。

しかし、データが見つからなかった場合、比較回数は要素数に正比例して増大するという欠点がある。

二分探索

プログラムの流れは以下の通りである。

1. データを昇順または、降順に並べ替える
2. そして、データが全体の前半分にあるか後ろ半分にあるかを判定する
3. 半分になったデータ群の中央と探しているデータを比較する
4. 2, 3を繰り返し、中央のデータが探しているデータに一致するか、データが見つからなかったときに探索を終了する

プログラムは、この流れで実行される。

利点としては、線形探索に比べて計算量を減らすことが出来るというものがある。

しかし、データを並べ替える手間が発生する欠点がある。

エラーが発生したときに、プログラムを調べる方法として、二分探索の考え方を使うとうまくいくことがある。例えば、ある関数の上半分を削除して結果を確認、次に下半分を削除して

結果を確認するというやり方がある。

1. 2. その他の探索手法について

木構造での探索

① 幅優先探索

探索を開始する独鈷炉から近いものをリストアップし、さらにそれぞれを細かく調べていく方法を幅優先探索という。

本を読むときに目次を観て全体を把握し、さらにそれぞれ章について概要を読み、さらに内容を読むという用に徐々に深く読んでいくようなイメージ
深さ優先探索に比べてメモリ使用量を抑えることが出来る。

② 深さ優先探索

目的の方に進めるだけ進んで、進めなくなったら戻する方法を深さ優先探索と言う。

再帰処理が使われることが多く、オセロや将棋などの対戦型のゲーム探索を行う場合には必須の探索方法である。

最短でたどり着けるものを一つだけ見つける場合には、身つかった時点で、処理を終了できるため、幅優先探索に比べて高速に動作する。

2. 線形探索について

for文を用いて、前からシンプルに探索する方法プログラムを記述した。

3. 二分探索について

半分になったモノを記述するというやり方で書いた。

他に二分探索を記述出来るやり方は特に無かった。

【結果】

```

3
開始位置 = 0, 中央位置 = 24終了位置 = 49
中央より左側にありそうです
開始位置 = 0, 中央位置 = 11終了位置 = 23
中央より右側にありそうです
開始位置 = 12, 中央位置 = 17終了位置 = 23
中央より右側にありそうです
開始位置 = 18, 中央位置 = 20終了位置 = 23
中央より左側にありそうです
開始位置 = 18, 中央位置 = 18終了位置 = 19
中央より右側にありそうです
開始位置 = 19, 中央位置 = 19終了位置 = 19
見つかりました
19

```

図1 プログラムの出力結果

【考察】

探索のやり方は、思ったよりも多いことがわかった。

その探索方法にもメリット、デメリットがあることを知った。

また、クラスメイトから教えてもらった、Javanの書き方を用いて、プログラムを記述することが出来た。Pythonと違い、import文を書かずとも、クラスを使うことを初めて知った。次からも用いるクラスを別に記述するというオブジェクト指向を用いて記述していきたい。

【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>