

アルゴリズムとデータ構造

演習第 10 回

サーチ 1 (二分探索)

データの中から、あるデータを探すことを探索といいます。ここでは二分探索（アルゴリズムC 第2巻 p.8）を用いて探索を行います。

問題 1 [印刷用 PostScript]

次のようなソートされたデータがある。

1 4 6 9 10 13 19 23 25 30

- (1) このデータから二分探索を用いて 9 を探索する過程を書きなさい。
- (2) このデータから内挿探索を用いて 9 を探索する過程を書きなさい。

二分探索、内挿探索はソートされているデータに対して行う探索方法です。

二分探索 (アルゴリズム C 第2巻 p.8)

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
4	8	10	13	16	17	19	20	23	25
↑				↑			↑		↑
l				x	$\frac{0+9}{2}$				r
4	8	10	13	16	17	19	20	23	25
					↑		↑		↑
					l		x	$\frac{5+9}{2}$	r

$$x = \frac{l+r}{2}$$

$$= l + \frac{1}{2}(r-l)$$

真ん中のデータが見つけたいデータかどうか調べます。 見つけたいデータが真ん中のデータより小さければ左側に対して、 大きければ右側に対して、同じことを繰り返します。

内挿探索 (アルゴリズムC 第2巻 p.12)

Diagram illustrating the binary search algorithm on a sorted array. The array is $a[0] \dots a[9]$ with values $4, 8, 10, 13, 16, 17, 19, 20, 23, 25$. The search process is shown in two steps.

Step 1: Initial range $l=0, r=9$. The middle element is calculated as $x = 0 + \frac{20 - 4}{25 - 4} (9 - 0) = 7$. The element at index 7 is 20, which is the target value.

Step 2: The range is updated to $l=7, r=9$. The middle element is calculated as $x = 7 + \frac{20 - 20}{25 - 20} (9 - 7) = 7$. The element at index 7 is 20, which is the target value.

二分探索は真ん中を調べていましたが、内挿探索では $\frac{v - a[l]}{a[r] - a[l]}$ の場所を調べます。これは二分探索とは違い、データの数値から見つけたいデータがどのあたりにあるか推測しています（二分探索は数値は見ない）。

この例ではどちらの探索も二回で見つけていますが、データが大きくなった場合は圧倒的に内挿探索の方が回数が少なく済みます。

問題 2

二分探索、内挿探索を用いてデータを探索するプログラムを作成しなさい。プログラムは以下の条件を満たすこと。（[ex10-2-skel.c](#)）

- 始めにデータをすべて表示する
- 二分探索、内挿探索それぞれに対して探索するキーを入力し、その結果を表示する

実行例：

```
% ./a.out
```

```
data:  1 4 6 9 10 13 19 23 25 30
```

```
Input key for binary search:  9
found.
```

```
Input key for interpolation search:  9
found.
```

```
% ./a.out
```

```
data:  1 4 6 9 10 13 19 23 25 30
```

```
Input key for binary search:  8
not found.
```

```
Input key for interpolation search:  8
not found.
```

データは、例えば次のように、配列を宣言するときに初期化しても良いです。ただし、これらの探索方法はデータがソートされていることが前提なので、ソートされた状態で初期化するようにしてください。

```
int a[N]={ 1, 4, 6, 9, 10, 13, 19, 23, 25, 30 };
```

内挿探索関数は、二分探索関数の $x=(l+r)/2$; の部分を $x=l+(v-a[l])*(r-l)/(a[r]-a[l])$; に変更すればできます（[アルゴリズム C 第2巻 p.12](#)）。

問題 3

二分探索、内挿探索において、探索終了までの比較回数を数え、表示するプログラムを作成しなさい。（[ex10-3-skel.c](#)）プログラムは以下の条件を満たすこと。

- データは乱数を用いて 10000 個作成する

- データはソートされていなくてはいけないので、 [演習第 8 回](#)または [演習第 9 回](#)で作成した関数を使用してソートする
- データは多いので、始めの 30 個だけ表示させる

また、問題 2 の内挿探索関数では、データによっては無限ループ（または segmentation fault になる）場合があるので、これも直しなさい

実行例：

% **./a.out**

data:

9	10	16	20	20	25	31	40	41
47	50	60	63	65	67	67	73	77
78	83	84	84	86	87	91	92	99

Input key for binary search: **99**

found.

Number of Comparison: 10

Input key for interpolation search: **99**

found.

Number of Comparison: 2

問題 2 の内挿探索関数は、分母（ $a[r]-a[l]$ ）が 0 になる場合や、分子（ $v-a[l]$ ）が負になる場合がまったく考慮されていません。 こうならないように改良してください。

Written by わかまつなおき