

アルゴリズムとデータ構造 プログラミング演習 二分探索

森 立平 mori@c.titech.ac.jp

2017 年 6 月 13 日

今日の目標

- 二分探索のプログラムを書けるようになる。
- 二分探索を使いこなせるようになる。

今日の主な課題 (提出締切は授業終了時)

1. 二分探索のプログラムを完成させる。
2. 二分探索を用いてより難しい問題を解く。

今日のワークフロー

1. この資料をよく読み二分探索の考え方及び、その C 言語プログラムについて理解する。
2. 5章に書いてある課題に取り組む。
3. 教員か TA に課題提出。

1 はじめに

ウェブブラウザで <https://github.com/ryuhei-mori/2017ad> にアクセスして「プログラミング演習のルール」を必ず読むこと。他にもこの資料や C 言語のプログラムが置いてある。

2 二分探索について

二分探索とはソートされている配列から、与えられた数 x 以上の最小の数及びそのインデックスをもとめるアルゴリズムである。例で説明する。次の長さ 12 のソートされている配列について、 $k = 17$ 以上の最小の数をもとめたいとする。

1	2	4	4	5	9	10	14	19	30	72	99
---	---	---	---	---	---	----	----	----	----	----	----

最初に配列の真ん中にある要素について $k = 17$ 以上かどうかチェックする。

1	2	4	4	5	9	10	14	19	30	72	99
---	---	---	---	---	---	----	----	----	----	----	----

9 は 17 よりも小さいので、この 9 よりも右にある要素からなる配列について $k = 17$ 以上の最小の数をもとめればよい。

1	2	4	4	5	9	10	14	19	30	72	99
---	---	---	---	---	---	----	----	----	----	----	----

残りの要素の中から真ん中の要素について $k = 17$ 以上かどうかチェックする。

1	2	4	4	5	19	10	14	19	30	72	99
--------------	--------------	--------------	--------------	--------------	---------------	----	----	----	----	----	----

19 は 17 以上なので、この 19 を含む 19 よりも左側にある要素からなる配列について $k = 17$ 以上の最小の数をもとめればよい。

1	2	4	4	5	19	10	14	19	30	72	99
--------------	--------------	--------------	--------------	--------------	---------------	----	----	----	---------------	---------------	---------------

残りの要素から真ん中の要素を調べる。

1	2	4	4	5	19	10	14	19	30	72	99
--------------	--------------	--------------	--------------	--------------	---------------	----	----	----	---------------	---------------	---------------

14 は 17 よりも小さいので、この 14 よりも右側だけを調べればよい。

1	2	4	4	5	19	10	14	19	30	72	99
--------------	--------------	--------------	--------------	--------------	---------------	---------------	---------------	----	---------------	---------------	---------------

19 だけが残ったので 19 が解である。

3 二分探索のプログラム

長さ n の配列について k 以上の最小の要素を出力する C 言語プログラムは次のようになる。ただし k 以上の要素が無い場合は、“NO” を出力する。

```
int lb, ub;
lb = -1;
ub = n;

while(ub - lb > 1) {
    int mid = (lb + ub) / 2;
    if(A[mid] >= k){
        ub = mid;
    }
    else{
        lb = mid;
    }
}

if(ub == n) {
    printf("NO\n");
}
else {
    printf("The answer is %d\n", A[ub]);
}
```

変数 lb と ub の意味は

- 変数 lb は $A[i] < k$ を確認した最大の i
- 変数 ub は $A[i] \geq k$ を確認した最小の i

である。便宜上 $A[-1]$ をマイナス無限、 $A[n]$ をプラス無限と解釈しているので、 $lb = -1$ 、 $ub = n$ と初期化する。実際にプログラムの中でアクセスされるのは $A[0]$ から $A[n-1]$ までなので心配はいらない。もしも $ub == lb + 1$ であれば、 $A[ub]$ が $A[i] \geq k$ を満たす最小の値である。 $ub - lb > 1$ の場合、while ループの中でまず $mid = (lb + ub) / 2$ を計算する (小数点以下切り捨てとなる)。このとき、 $lb < mid \ \&\& \ mid < ub$ が満たされることに注意する。 $A[mid] \geq k$

が満たされていれば $ub = mid$ に更新し、満たされていなければ $lb = mid$ と更新する。ここで、 $ub - lb$ は必ず減少するので無限ループに陥いることはない。前節の例に対して上記のプログラムお実行した場合に lb と ub がどのように更新されるかを下の図で示す。

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
$-\infty$	1	2	4	4	5	9	10	14	19	30	72	99	$+\infty$

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
$-\infty$	1	2	4	4	5	9	10	14	19	30	72	99	$+\infty$

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
$-\infty$	1	2	4	4	5	9	10	14	19	30	72	99	$+\infty$

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
$-\infty$	1	2	4	4	5	9	10	14	19	30	72	99	$+\infty$

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
$-\infty$	1	2	4	4	5	9	10	14	19	30	72	99	$+\infty$

最終的に $ub == 8$ となる。

4 より高度な二分探索の使い方

二分探索はソートされた配列から値を探すという目的以外にも使用することができる。ソートされた配列の代わりに単調な二値関数 $p: \{0, 1, 2, \dots, n-1\} \rightarrow \{0, 1\}$ について考えよう。ここで p が単調というのは $p(i) \geq p(i-1)$ が $i = 1, 2, \dots, n-1$ について満たされるという意味である。このような p について

$p(i) = 1$ を満たす最小の $i \in \{0, 1, 2, \dots, n-1\}$ をもとめよ

という問題を二分探索で解くことができる。具体的には前節のプログラムのうち 5 行目からの 9 行を次のように変更すればよい。

```
while(ub - lb > 1) {
    int mid = (lb + ub) / 2;
    if(p(mid)){
        ub = mid;
    }
    else{
        lb = mid;
    }
}
```

実際に変更したのはこの中の 3 行目の if 文の条件だけである。この考え方を利用すると、上記の問題について非常に大きな n に対しても現実的な時間で解くことができる。

5 課題

以下の問題についてプログラムを書け。A4 の紙に

- テスト用の入力 <https://github.com/ryuhei-mori/2017ad/tree/master/bsearch/src/inputs> に対する出力
- 問題 5.2, 5.3 における関数 p の定義と p のプログラムコード

を書いて提出すること。入力がテキストファイル `input.txt` に保存されているとき、`./a.out < input.txt` で入力を読み込ませることができる。

5.1 配列の二分探索 ★

問題文

単調非減少数列 a_0, a_1, \dots, a_{n-1} と整数 k について、 $a_x \geq k$ となる最小の整数 x を求めよ。ただし、そのような x が存在しないときは $x = n$ とする。

入力制約

$$\begin{aligned} 1 &\leq n \leq 10^5 \\ 1 &\leq k \leq 10^9 \\ 1 &\leq a_i \leq 10^9, \quad i \in \{0, 1, \dots, n-1\} \\ a_i &\leq a_{i+1}, \quad i \in \{0, 1, \dots, n-2\} \end{aligned}$$

入力

$n \ k$
 $a_0 \ a_1 \ \dots \ a_{n-1}$

出力

x

サンプル 1

入力:
12 17
1 2 4 4 5 9 10 14 19 30 72 99
出力:
8

サンプル 2

入力:
12 3
1 2 4 4 5 9 10 14 19 30 72 99
出力:
2

サンプル 3

入力:
1 10
9
出力:
1

5.2 りんご狩り ★★

問題文

りんご狩りに n 人集まった。各 $i = 1, 2, \dots, n$ について i 番目の人は a_i 個のりんごを収穫した。合計 k 個のりんごバッグ (サイズは全て同じ) を配ると、全ての人はりんごを持ち帰ることができた。りんごバッグに入れることができるりんごの個数としてあり得るもののうち最小値 x をもとめよ。

入力制約

$$\begin{aligned} 1 \leq n &\leq 10^4 \\ 1 \leq a_i &\leq 10^5, \quad i \in \{1, 2, \dots, n\} \\ n \leq k &\leq 10^4 \end{aligned}$$

入力

$$\begin{aligned} n \ k \\ a_1 \ a_2 \ \dots \ a_n \end{aligned}$$

出力

$$x$$

サンプル 1

入力:

$$\begin{aligned} 5 \ 7 \\ 1 \ 2 \ 3 \ 4 \ 5 \end{aligned}$$

出力:

$$3$$

サンプル 2

入力:

$$\begin{aligned} 5 \ 5 \\ 10 \ 20 \ 30 \ 40 \ 1000 \end{aligned}$$

出力:

$$1000$$

サンプル 3

入力:

$$\begin{aligned} 7 \ 1000 \\ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \end{aligned}$$

出力:

$$1$$

5.3 槍作り ★★

問題文

長さ m の木から長さ x の木製の槍が $\lfloor m/x \rfloor$ 本作れる。今 n 本の木があり、各木の長さは a_1, a_2, \dots, a_n である。これら木を切り同じ長さの木製の槍を k 本作りたい。作ることができる槍の長さの最大値 x をもとめよ。

入力制約

$1 \leq n \leq 10^4$
 $1 \leq a_i \leq 10^5, \quad i \in \{1, 2, \dots, n\}$
 $1 \leq k \leq 10^4$

入力

$n \ k$
 $a_1 \ a_2 \ \dots \ a_n$

出力

x

サンプル 1

入力:

5 7
1 2 3 4 5

出力:

1

サンプル 2

入力:

5 5
10 20 30 40 1000

出力:

200

サンプル 3

入力:

7 1000
1 2 3 4 5 6 7

出力:

0