

# アルゴリズムとデータ構造

## 二分探索の応用

森 立平

mori@c.titech.ac.jp

2018 年 6 月 19 日

今日のメッセージ

- 二分探索には様々な応用がある
- C 言語は簡単

今日の演習の目標

- 二分探索を応用したプログラムを書けるようになる

今日の主な演習課題 (提出締切は来週火曜日正午 (12:00))

1. 二分探索を応用したプログラムを書く

今日の演習時間のワークフロー

1. Algorithms-Datastructures/docs/git.pdf を見て、git の使い方を覚える
2. Git の資料にあるように、fork, clone, ファイルを編集, commit, push, pull request という流れで課題を提出する。

## 1 二分探索の C 言語プログラム

前回説明したように単調な関数  $p: \mathbb{Z} \rightarrow \{0, 1\}$  について、 $p(z) = 1$  となる最小の  $z \in \mathbb{Z}$  を見つける二分探索のプログラムは次のようになる。

```
int lb, ub;
lb = 「絶対に p(lb) == 0 となる値」;
ub = 「絶対に p(ub) == 1 となる値」;
while(ub - lb > 1) {
    int m = (lb + ub) / 2;
    if(p(m)){
        ub = m;
    }
    else {
        lb = m;
    }
}
/* ここで ub が p(ub) == 1 となる最小の値
   lb が p(lb) == 0 となる最大の値となっている */
```

例えば入力として整数  $n \geq 1$  をとり、 $\lceil \sqrt{n} \rceil$  を出力するプログラムは次のようになる。

```

#include <stdio.h>

int n;

int p(int m){
    return (long long int) m * m >= n;
}

int main(){
    int lb, ub;
    scanf("%d", &n);
    lb = 0;
    ub = n;
    while(ub - lb > 1) {
        int m = (lb + ub) / 2;
        if(p(m)){
            ub = m;
        }
        else {
            lb = m;
        }
    }
    printf("%d\n", ub);
    return 0;
}

```

- #include <stdio.h> は毎回書くおまじない。入出力関係のライブラリを使うためのヘッダファイルの取り込み。これが無いと scanf や printf が使えない。
- int n; のような関数の外側にある変数は**グローバル変数**と呼ばれ、プログラムのどこからでもアクセスできる。
- int main() は **main 関数**という特別な関数で、プログラムを実行するとこの main 関数が実行される。
- int lb, ub; のような関数の内側にある変数は**ローカル変数**と呼ばれ、その関数の内側からだけアクセスできる。
- scanf("%d", &n); は**標準入力から整数を読み込み** n に代入する命令文である。標準入力から整数を 2 つ読み込んで int 型の変数 a と b に代入したい場合は scanf("%d%d", &a, &b); とする。
- printf("%d\n", &n); は標準出力へ整数 n と改行コード \n を出力する。
- return 0; は main 関数の最後に書いておこう (書かなくてもエラーが起きる訳ではないが)。これは「プログラムが正常に終了した」ということを表わしている。エラーで終了するときは return 1; など 0 以外の値を返すようにする。この授業ではエラーによる終了は扱わないので、main 関数の最後に return 0; を書くことを覚えておけばよい。

他によくある処理として、最初に数列の長さ  $n$  を、その後に数列  $A_0, \dots, A_{n-1}$  を読み込みたい場合は

```

int i, n;
scanf("%d", &n);
for(i = 0; i < n; i++){
    scanf("%d", &A[i]);
}

```

とする。ここで

```
for(AAA; BBB; CCC){  
    DDD  
}
```

は

```
AAA;  
while(BBB){  
    DDD  
    CCC;  
}
```

と等価である。ここで、AAA, BBB, CCC はそれぞれ一つの文であり、DDD は文の列であるとする。

標準入力とは特定のファイルなどではなくプログラムが受け付ける入力であり、scanf を用いたプログラムを実行すると入力受け付け状態になる。入力と改行 (Enter キー) を入力すると、scanf により読み取られて残りの処理が実行される。毎回入力するのが面倒くさい場合はターミナルで `echo 100 | ./a.out` とすれば `./a.out` としてから 100 を入力して Enter を押すのと同じことになる。また、ファイル `file.txt` に入力を記入しておいてから、`./a.out < file.txt` としてもよい。

## 2 より高度な二分探索

二分探索とは、単調な関数  $p: \mathbb{Z} \rightarrow \{0, 1\}$  について、 $p(x) = 1$  となる最小の  $x \in \mathbb{Z}$  を見つけるアルゴリズムであった。この二分探索は適用範囲が広く、様々な応用がある。例えば、今回の課題 (後述) の「りんご狩り」では

$$p(x) := \begin{cases} 1, & \text{if } x \text{ 個のりんごが入るりんごバッグを合計 } k \text{ 個配ると全員がりんごを持って帰れる} \\ 0, & \text{otherwise.} \end{cases}$$

と定義すれば  $p$  は単調性を満たし、 $p(x) = 1$  となる最小の  $x$  が求める解となる。この  $p(x)$  は効率的に計算することができるので、この問題を十分効率的に解くことができる。

### 3 課題

#### 3.1 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

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

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

### 3.2 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

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

入力

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

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:

$$1$$

### 3.3 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。これらの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

$1 \leq n \leq 10^5$   
 $1 \leq a_i \leq 10^9, \quad i \in \{1, 2, \dots, n\}$   
 $1 \leq k \leq 10^5$

入力

$n \ k$   
 $a_1 \ a_2 \ \cdots \ a_n$

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

### 3.4 仕事配分 ★★

問題文

$n$  時間の仕事を  $k$  人の人に分配したい。それぞれの人は連続した何時間かを働き、また異なる人が同時に働くことはないものとする。各時間の仕事量を整数列  $a_1, a_2, \dots, a_n$  とするとき、一番仕事量が多い人の仕事量を最小化するように仕事を配分するとする。そのような仕事配分をしたとき、一番仕事量が多い人の仕事量  $x$  をもとめよ。

入力制約

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

入力

$n \ k$   
 $a_1 \ a_2 \ \cdots \ a_n$

出力

$x$

サンプル 1

入力:

3 2  
1 2 4

出力:

4

サンプル 2

入力:

3 2  
1 4 2

出力:

5

サンプル 3

入力:

5 5  
10 20 30 40 1000

出力:

1000