# Cプログラミング I 演習

# 探索

# ファイルの中から「ある数」を見つける

例えば、100万個のデータから、ある特定の「数」をみつけよう
どうしたらよいだろうか？

すぐに考えられるのは、一つ一つデータを見て、目的の「数」かどうか照合していく

これが「逐次探索」(sequential search)

人間がやるのは大変だけれど、コンピュータならできる！

# 逐次探索

- 前提
  ファイルの中のデータが配列 array に入っている:
  array[0]〜array[n-1]  --- つまりデータの個数はn
  探索すべき数が変数targetの値になっている
- 手順:

1. for  i  in 0...n-1
2.  　if array[i] == target
3.  　　return i  /* targetが存在したインデックス */
4. return -1  /* -1 は「存在しない」ことを意味する */

# 課題1.

data.txtというファイル（データの個数は未知とする）から次の数があるかどうか調べるプログラムをかき、その結果を答えよ。

(1) 500

(2) 801

# 計算の手間

この方法で、100万個のデータから特定の数を見つけるにはどのくらいデータの照合が行われるか、考えてみよう

最悪の場合：

期待値(データの中にtargetがあるとして)：

# データが整理されていれば…

図書館を考えてみよう。数百万冊の中からある特定の本を容易に探すことができるのは、本が整理されているからである。
これと同じように、データが整理されていれば、特定の数を探すのはやりやすくなる

ここで「データの整理」には何が使えるだろうか？

# そうだ、ソートだ

数の整理には、ソートという方法があった
（すでに、選択ソート、クイックソートなどの方法を学んだ）

数のソート(sorting)とは、（昇順の場合）以下のように決まった順番に数を並べたものであった:

15，35, 40, 108，112, …

# データがソートされていれば

ソートされていれば、「二分探索」(binary search)という方法が取れる。

これは大雑把に言うと、n個のデータに対し
「ちょうど真ん中」のデータと target を比べる
　そのデータがtargetよりも小さければ、もっと大きい方を「二分探索」で探す
　そのデータが大きければ、もっと小さい方を「二分探索」で探すという方法。要するに探索範囲を半分ずつに絞り込む方法

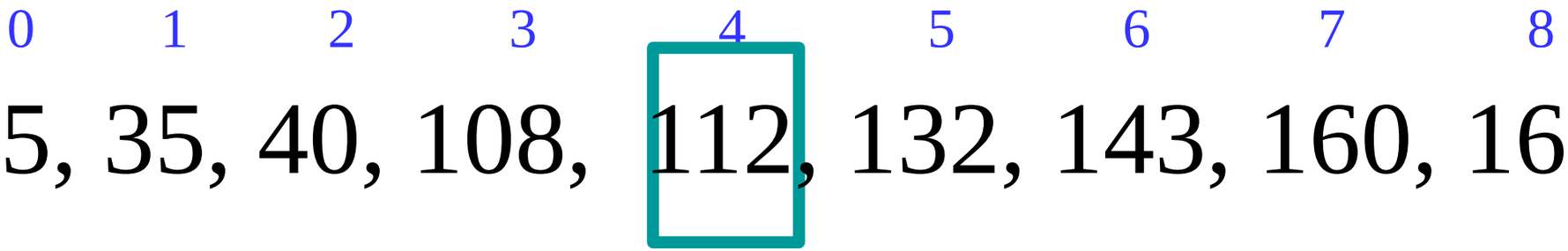
# 二分探索の具体的なイメージ

9個のデータが次のように並んでいるとする：

そこで108という数があるかどうかを「二分探索」で探してみよう

15, 35, 40, 108,  112, 132, 143, 160, 167

ここで活躍するのがインデックス
探索範囲を s から e とする(開始時点では 0～8)
その「ちょうど半分」は、(s+e)/2 で求まる




112は108よりも大きいので、探索範囲は、この左半分：0から3となる

# 二分探索のアルゴリズム

前提
ファイルの中のデータが昇順で配列 array に入っている:
　　array[0]～array[n-1] --- つまりデータの個数はn
探索すべき数が変数targetの値になっている
関数名を bSearch, 引数を array, s, eとする。
　探索範囲の左端を表す変数 s　　初期値：0
　探索範囲の右端を表す変数 e　　初期値：n-1
手順

```
1.   if (s > e ) return -1                  // 見つからなかった
2.   m ← (s+e)/2                           //  ちょうど真ん中のインデックス
3.   if (array[m] == target)               // targetが真ん中の要素に一致した場合
          then      return m
     else if (array[m] > target)           // 真ん中の要素より小さい場合
               then return bSearch(array, s, m-1)      // arrayの小さい方を探索する
     else  return bSearch(array, m+1, e)   // それ以外(targeが真ん中の要素より大きい場合)
                                                  // arrayの大きい方を探索する
```

# 試してみよう、書いてみよう

以下のデータで、試してみよう

(1) 108を探索する　　(2) 150を探索する　　(3) 32を探索する

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 15, | 35, | 40, | 108, | 112, | 132, | 143, | 160, | 167 |

動きに納得がいったら、プログラムを書いてみよう