

アルゴリズムとデータ構造

第3回 二分探索法と計算量

第3回のキーワード

2

アルゴリズム関係

- 2分探索 (binary search)
- $O(\log n)$
- 再帰による2分探索

Java関係

- `compareTo`

この資料は図解が中心なので、文章は説明不足の部分もあります。
本講義の学修では、講義内容や演習課題もよく読んで理解してください。

もっと速い探索方法はないの？

3

- 前回紹介「線形探索」
 - ▣ 先頭から順番に調べていくしかないのか？
 - ▣ コンピュータなら、1000人の名前がバラバラに並んだ名簿からでも一瞬で検索できるけど...

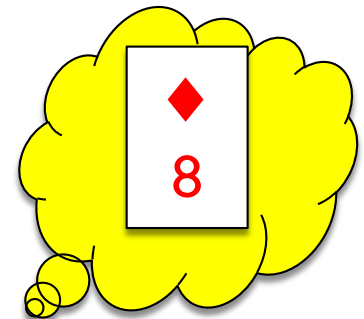
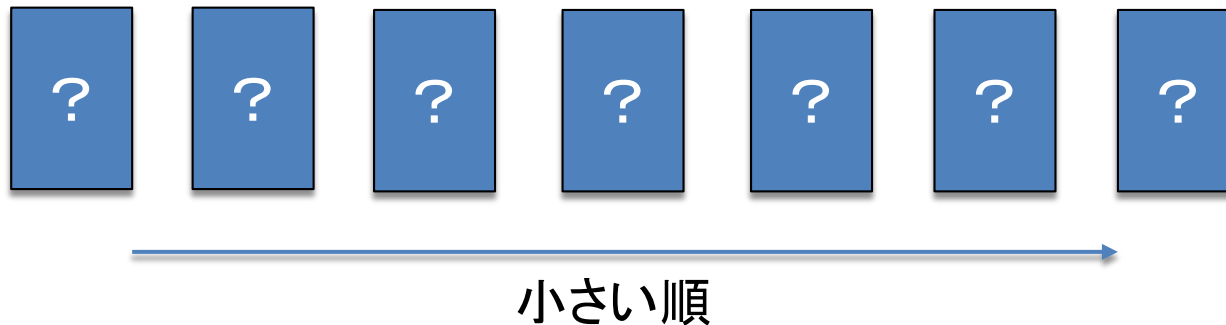
- データ構造を工夫する
 - ▣ もっと効率的に探し出すためには、どうすればいいか？
 - ▣ **世の中の情報**は、どのように整理してあるだろうか？

- 探索の高速化戦略
 - ▣ 整列しておく ⇒ 今回説明
 - ▣ 分類しておく ⇒ 似たようなことは次回以降説明

2分探索の考え方

4

- トランプにたとえると...
 - 裏返しのカードが、数の小さい順に並んでいる
 - この中で探したいカードはどこにあるか？

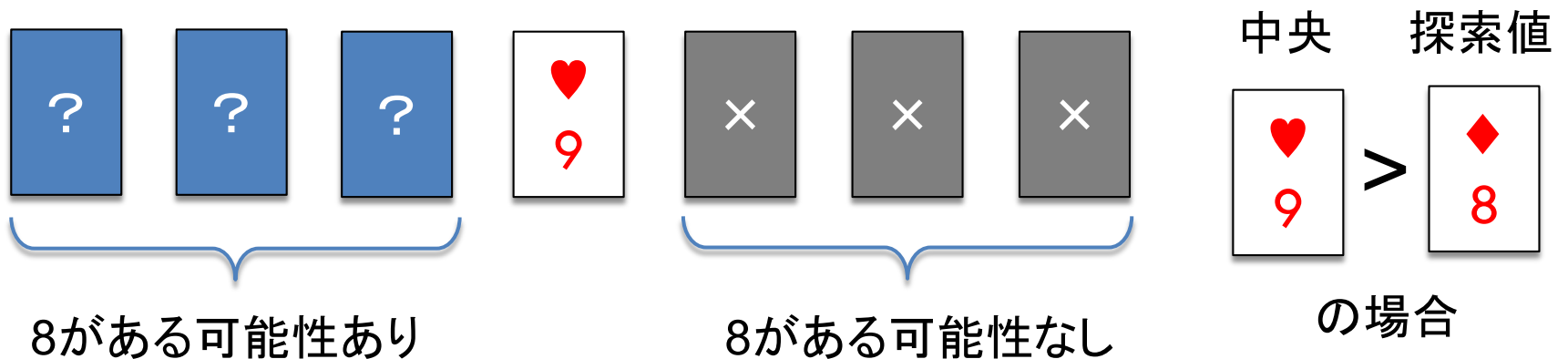
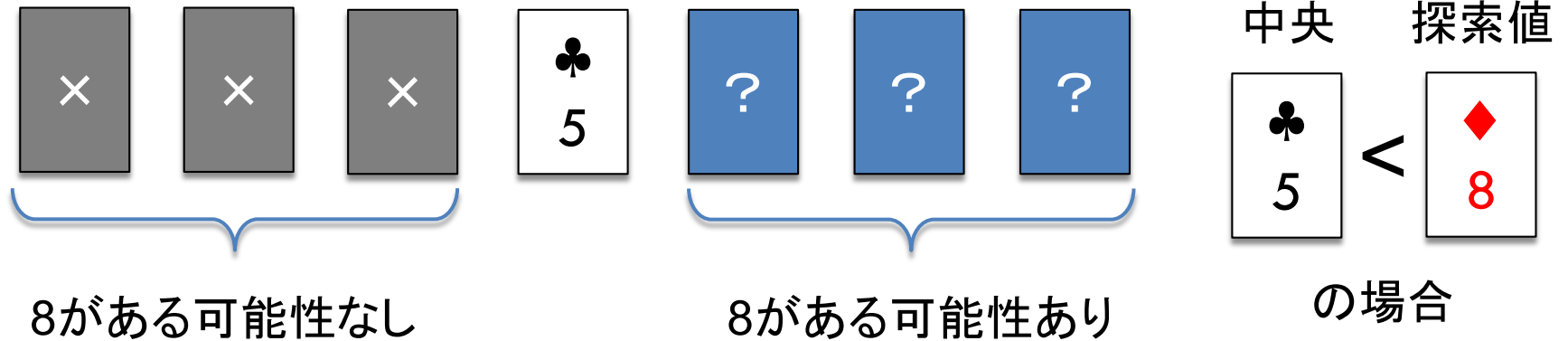


- 戦略を考えてみよう
 - まず、真ん中のカードを開けると何がわかるか？
 - 候補が半分ずつに減っていく(半分→4分の1→8分の1...)

2分探索の考え方

5

- 中央を開けると探索範囲を半分に絞り込める



2分探索のアルゴリズム

6

□ データ

- 配列の要素は、小さい順(または大きい順)に並べておく
- 探索する値をkeyとする

これが高速化の
ポイント！

□ アルゴリズム

- 配列(探索範囲)の中央にある値とkeyを比較する
- もし両者が等しければ、発見したのでその位置を返す
- もしkeyの方が小さければ、探索範囲を前半分にせばめる
- もしkeyの方が大きければ、探索範囲を後半分にせばめる
- 以上の手順を、探索範囲に要素がなくなるまで繰り返す
- 探索範囲に要素がなくなったら、keyは含まれていない

確認問題

7

□ 線形探索と2分探索の比較

- 下記の配列 α から「線形探索」で「7」を探るとき、発見するまでに比較する要素の値を順に並べ、比較回数を述べよ
- 【注意】プログラムを作るのではなく、自分が(人間が)数字を順に書き出す

α	6	8	2	3	5	9	1	7	4	0
----------	---	---	---	---	---	---	---	---	---	---

- 配列 α の内容を「2分探索」が適用できるように変更せよ

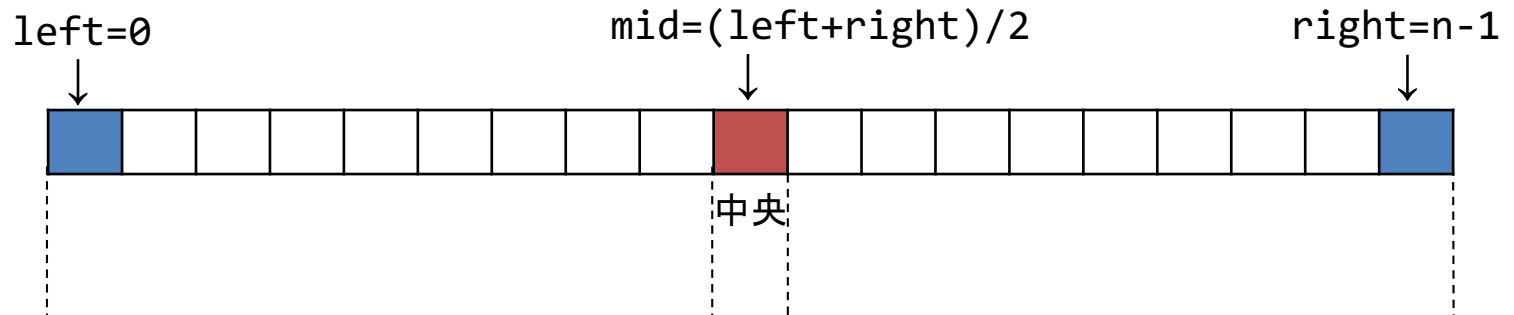
α										
----------	--	--	--	--	--	--	--	--	--	--

- その配列から「2分探索」で「7」を探るとき、発見するまでに比較する要素の値を順に並べ、比較回数を述べよ

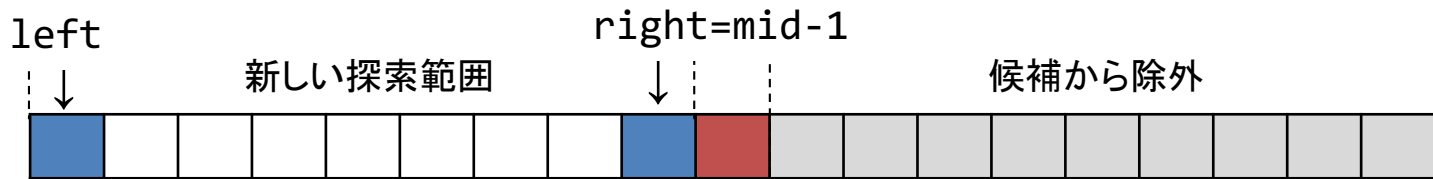
探索範囲のせばめ方

8

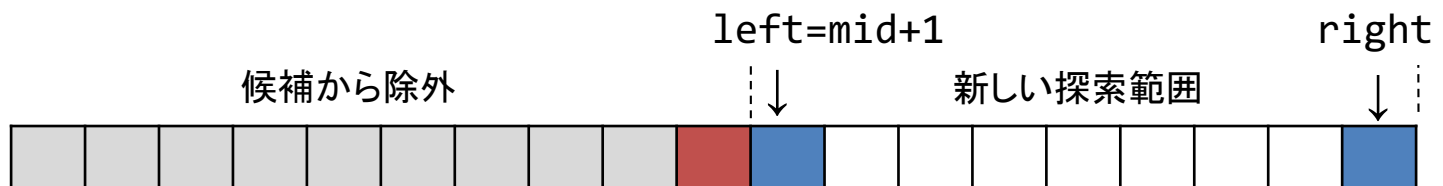
探索範囲の左端と右端を表す変数を使う



左半分にせばめるときは、右端を動かす



右半分にせばめるときは、左端を動かす



確認問題

9

- 必要な処理を追加してプログラムを完成させよ

```
public static int binarySearch(int key, int[] data) {  
  
    int left = 0; // 探索範囲の左端  
    int right = data.length - 1; // 探索範囲の右端  
  
    while (left <= right) { // 探索範囲が1個以上である間繰り返す  
        int mid =  
  
        if (key == data[mid]) {  
  
        } else if (key < data[mid]) {  
  
        } else {  
  
        }  
    }  
    return -1; // 発見できなかった場合は-1を返す  
}
```