

高等学校「情報I」対策講座

プログラミング

線形探索, 二分探索

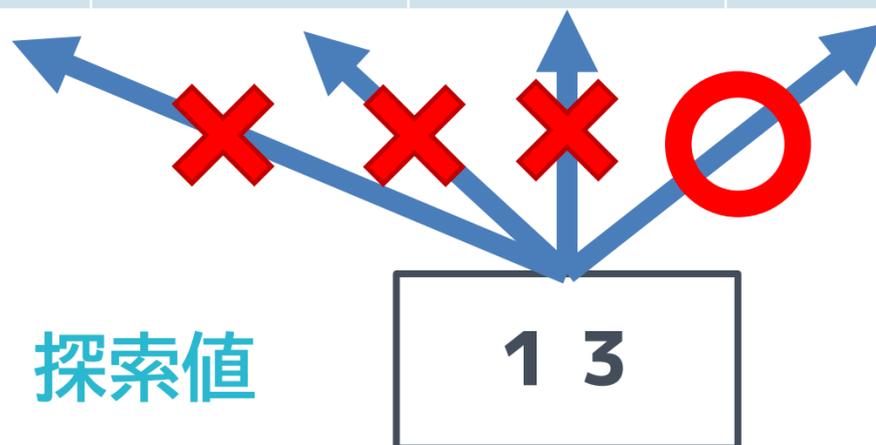


# 線形探索



配列の要素を先頭から順に見ながら，探索値を探す方法

添え字	0	1	2	3	4	5
要素	20	15	8	13	19	5



# 線形探索



配列の要素を先頭から順に見ながら，探索値を探す方法

```
a = [16, 23, 31, 48, 52, 65, 79, 84, 97]
n = len(a)
print('番号 データ')
for i in range(0, n, 1):
    print(i, ' ', a[i])
print("")
s = int(input('探索値の入力')) # ①
for i in range(0, n, 1):
    if a[i] == s:
        print(s, 'は', i, 'に存在')
        break
```

iの値	a[i]の値
0	16
ア	イ
ウ	エ
オ	カ
キ	ク
ケ	コ
サ	シ

プログラムにおいて、  
①の行が実行された時に探索値として「65」を入力した。  
この時「65」が見つかるまでの変数*i*および*a[i]*の値を、表にうめなさい。  
また、「65」が見つかった後の*i*の値、*a[i]*の値には「-」を記入しなさい。

# 線形探索



## 配列の要素を先頭から順に見ながら、探索値を探す方法

```
a = [16, 23, 31, 48, 52, 65, 79, 84, 97]
n = len(a)
print('番号 データ')
for i in range(0, n, 1):
    print(i, ' ', a[i])
print("")
s = int(input('探索値の入力')) # ①
for i in range(0, n, 1):
    if a[i] == s:
        print(s, 'は', i, 'に存在')
        break
```

iの値	a[i]の値
0	16
ア	イ
ウ	エ
オ	カ
キ	ク
ケ	コ
サ	シ

プログラムにおいて、  
①の行が実行された時に探索値として「65」を入力した。  
この時「65」が見つかるまでの変数*i*および*a[i]*の値を、  
表にうめなさい。  
また、「65」が見つかった後の*i*の値、  
*a[i]*の値には「-」を記入しなさい。

# 線形探索



## 配列の要素を先頭から順に見ながら、探索値を探す方法

```
a = [16, 23, 31, 48, 52, 65, 79, 84, 97]
n = len(a)
print('番号 データ')
for i in range(0, n, 1):
    print(i, ' ', a[i])
print("")
s = int(input('探索値の入力')) # ①
for i in range(0, n, 1):
    if a[i] == s:
        print(s, 'は', i, 'に存在')
        break
```

iの値	a[i]の値
0	16
1	23
2	31
3	48
4	52
5	65
-	-

プログラムにおいて、  
①の行が実行された時に探索値として「65」を入力した。  
この時「65」が見つかるまでの変数*i*および*a[i]*の値を、表にうめなさい。  
また、「65」が見つかった後の*i*の値、*a[i]*の値には「-」を記入しなさい。

# 二分探索



配列の探索範囲を半分ずつ狭めながら，探索値を探す方法

添え字	0	1	2	3	4	5
要素	5	8	13	15	19	20

探索値

13

# 二分探索



配列の探索範囲を半分ずつ狭めながら，探索値を探す方法

添え字	0	1	2	3	4	5
要素	5	8	13	15	19	20

探索値

13

添え字	0	1	2
要素	5	8	13

右半分は調べない！

# 二分探索



配列の探索範囲を半分ずつ狭めながら，探索値を探す方法

```
a = [16, 23, 31, 48, 52, 65, 79, 84, 97]
```

```
n = len(a)
```

```
print('番号 データ')
```

```
for i in range(0, n, 1):
```

```
    print(i, ' ', a[i])
```

```
s = int(input('探索値の入力')) # ①
```

```
i = 0
```

```
j = ア
```

```
イ i <= j:
```

```
    m = int(ウ)
```

```
    if エ == s:
```

```
        print(s, 'は', m, 'に存在')
```

```
        break
```

```
    if a[m] > s:
```

```
        オ
```

```
    else:
```

```
        カ
```

プログラムにおいて、  
①の行が実行された時に探索値として「65」を入力した。  
この時「65」が見つかるまでの変数*i*および*a[i]*の値を、表にうめなさい。  
また、「65」が見つかった後の*i*の値、*a[i]*の値には「-」を記入しなさい。

# 二分探索



配列の探索範囲を半分ずつ狭めながら，探索値を探す方法

```
a = [16, 23, 31, 48, 52, 65, 79, 84, 97]
```

```
n = len(a)
```

```
print('番号 データ')
```

```
for i in range(0, n, 1):
```

```
    print(i, ' ', a[i])
```

```
s = int(input('探索値の入力')) # ①
```

```
i = 0
```

```
j = n - 1
```

```
while i <= j:
```

```
    m = int((i + j) / 2)
```

```
    if a[m] == s:
```

```
        print(s, 'は', m, 'に存在')
```

```
        break
```

```
    if a[m] > s:
```

```
        i = m - 1
```

```
    else:
```

```
        j = m + 1
```

# 二分探索



配列の探索範囲を半分ずつ狭めながら，探索値を探す方法

```
a = [16, 23, 31, 48, 52, 65, 79, 84, 97]
```

```
n = len(a)
```

```
print('番号 データ')
```

```
for i in range(0, n, 1):
```

```
    print(i, ' ', a[i])
```

```
s = int(input('探索値の入力')) # ①
```

```
i = 0
```

```
j = n - 1
```

```
while i <= j:
```

```
    m = int((i + j) / 2)
```

```
    if a[m] == s:
```

```
        print(s, 'は', m, 'に存在')
```

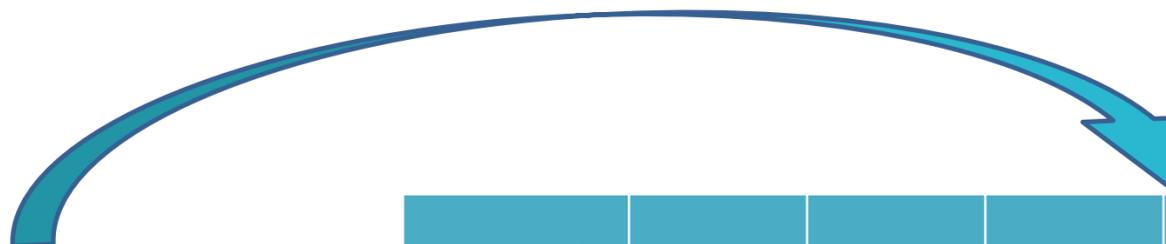
```
        break
```

```
    if a[m] > s:
```

```
        i = m - 1
```

```
    else:
```

```
        j = m + 1
```



添え字	0	1	2	3	4	5
要素	5	8	13	15	19	20

■は，何がはいるでしょうか

# 二分探索



配列の探索範囲を半分ずつ狭めながら，探索値を探す方法

```
a = [16, 23, 31, 48, 52, 65, 79, 84, 97]
```

```
n = len(a)
```

```
print('番号 データ')
```

```
for i in range(0, n, 1):
```

```
    print(i, ' ', a[i])
```

```
s = int(input('探索値の入力')) # ①
```

```
i = 0
```

```
j = n - 1
```

```
while i <= j:
```

```
    m = int((i + j) / 2)
```

```
    if a[m] == s:
```

```
        print(s, 'は', m, 'に存在')
```

```
        break
```

```
    if a[m] > s:
```

```
        j = m - 1
```

```
    else:
```

```
        i = m + 1
```

添え字	0	1	2	3	4	5
要素	5	8	13	15	19	20

j は，調べる範囲の末項  
i は，調べる範囲の初項

# 線形探索と二分探索



## メリットとデメリットを考えよう

探索とデータの並べかえに関する次の記述のうち、正しいものを一つ選べ。

- ① 線形探索はデータが順番に並んでいなくてもよいが、二分探索はデータが昇順または降順に並んでいる必要がある。
- ② 線形探索はデータが昇順または降順に並んでいる必要があるが、二分探索はデータが順番に並んでいなくてもよい。
- ③ 線形探索も二分探索もデータが昇順または降順に並んでいる必要がある。
- ④ 線形探索も二分探索もデータが昇順または降順に並んでいなくてもよい。

# 線形探索と二分探索



## メリットとデメリットを考えよう

探索とデータの並べかえに関する次の記述のうち、正しいものを一つ選べ。

- ① 線形探索はデータが順番に並んでいなくてもよいが、二分探索はデータが昇順または降順に並んでいる必要がある。
- ② 線形探索はデータが昇順または降順に並んでいる必要があるが、二分探索はデータが順番に並んでいなくてもよい。
- ③ 線形探索も二分探索もデータが昇順または降順に並んでいる必要がある。
- ④ 線形探索も二分探索もデータが昇順または降順に並んでいなくてもよい。

# 線形探索と二分探索



メリットとデメリットを考えよう

データ数が100のとき、線形探索と二分探索データのそれぞれの場合で、平均探索回数と最大探索回数を求めよ。

探索の種類	平均探索回数	最大探索回数
線形探索		
二分探索		

# 線形探索と二分探索



メリットとデメリットを考えよう

データ数が100のとき、線形探索と二分探索データのそれぞれの場合で、平均探索回数と最大探索回数を求めよ。

探索の種類	平均探索回数	最大探索回数
線形探索	50.5	100
二分探索	6	7

# 線形探索と二分探索

✔ メリットとデメリットを考えよう

データ数が100のとき

探索の種類	平均探索回数	最大探索回数
線形探索	50.5	100
二分探索	6	7

データ数がnのとき

探索の種類	平均探索回数	最大探索回数
線形探索	$(n+1) \div 2$	n
二分探索	$\log_2 n$	$(\log_2 n) + 1$

とても効率がよい！

# 整列（ソート）プログラム



データを大きい順や小さい順に並び替える

## 交換法

添え字	0	1	2	3
要素	13	20	7	5

添え字	0	1	2	3
要素	13	7	20	5

添え字	0	1	2	3
要素	13	7	5	20

後ろから  
決定する

前後の交換

前後の交換

## 選択法

添え字	0	1	2	3
要素	13	20	7	5

添え字	0	1	2	3
要素	7	20	13	5

添え字	0	1	2	3
要素	5	20	13	7

前から  
決定する

先頭との交換

先頭との交換