

【探索(線形探索)】

#探索:線形探索

```
Data = [3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27]
sagasu = int(input("探索する数 : "))
yousosu = len(Data)
flag = 0
kaisu = 1
while flag == 0 and kaisu  yousosu :
    if sagasu == Data[  ] :
        flag = 
    else:
        kaisu = 
if flag ==  :
    print(sagasu, "を", kaisu, "番目に発見 !")
else:
    print(sagasu, "はありませんでした !")
```

#見つかった時は「1」に変更
#比較した回数

» 探索する数 : 17

#例として「17」を入力

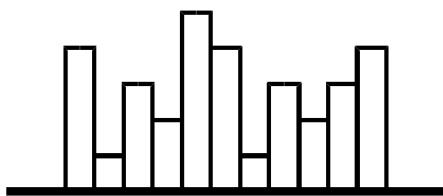
17 を 8 番目に発見 !

探索する数 : 20

#例として「20」を入力

20 はありませんでした !

【整列(バブルソート)】



$n = 13$
 $m = 12 \sim 1 \Rightarrow n-1 \sim 1$
 $k = 1 \sim m \Rightarrow 1 \sim m$

#整列:バブルソート

```

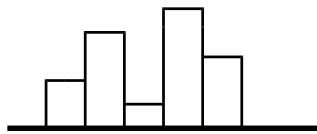
Data = [7, 3, 8, 0, 2, 3, 6, 9, 3, 7, 4, 1, 5]
n = len(Data)
for m in range(□, □, □):
    for k in range(□, □, □):
        if Data[ k-1 ] > Data[ k ]:
            temp = Data[ k-1 ]
            Data[ □ ] = Data[ □ ]
            Data[ □ ] = temp
print(Data)

```

#mは...
#kは...
#データの入れ替えブロック

» [0, 1, 2, 3, 3, 3, 4, 5, 6, 7, 7, 8, 9]

【整列(選択ソート)】



$n = 13$
 $m = 0 \sim 11 \Rightarrow 0 \sim n-2$
 $k = m+1 \sim 12 \Rightarrow m+1 \sim n-1$

#整列:選択ソート

```

Data = [7, 3, 8, 0, 2, 3, 6, 9, 3, 7, 4, 1, 5]
n = len(Data)
for m in range(0, n-2, 1):
    index_min = □
    data_min = □
    for k in range(m+1, n, 1):
        if Data[ □ ] < data_min:
            index_min = □
            data_min = Data[ □ ]
    Data[ □ ] = Data[ □ ]
    Data[ □ ] = data_min
print(Data)

```

#mは...
#kは...
#より小さいか否かの判定
#インデックス番号の記憶
#より小さい値の記憶
#最小値の入れ替えブロック

» [0, 1, 2, 3, 3, 3, 4, 5, 6, 7, 7, 8, 9]