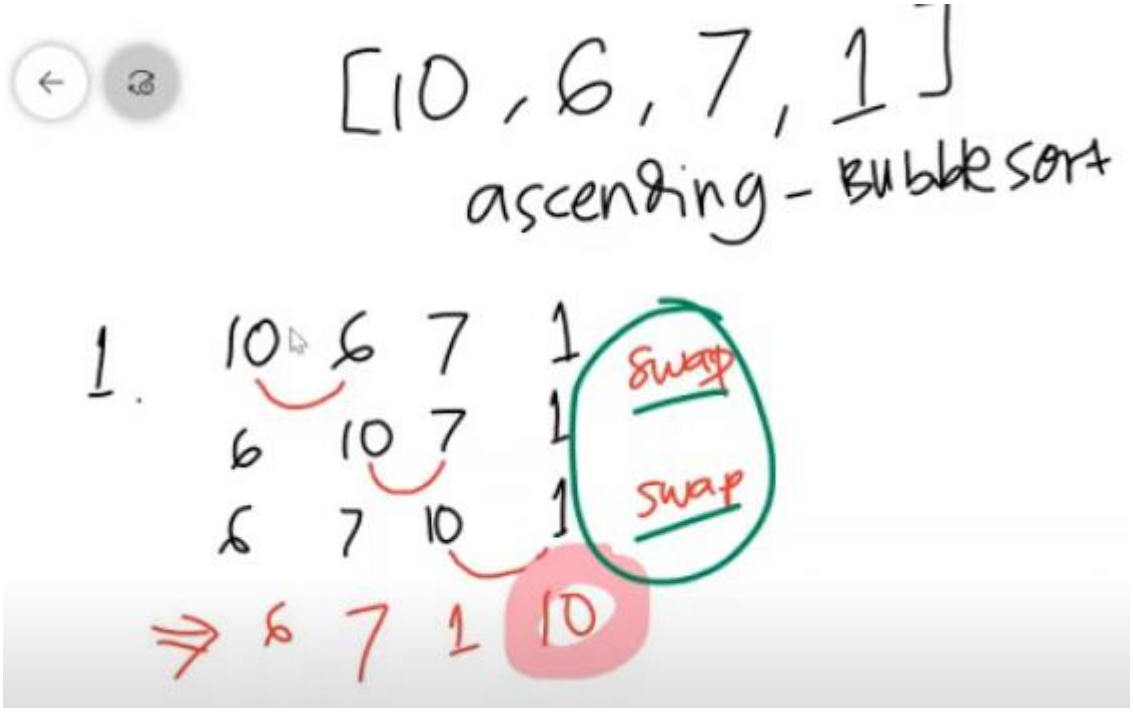
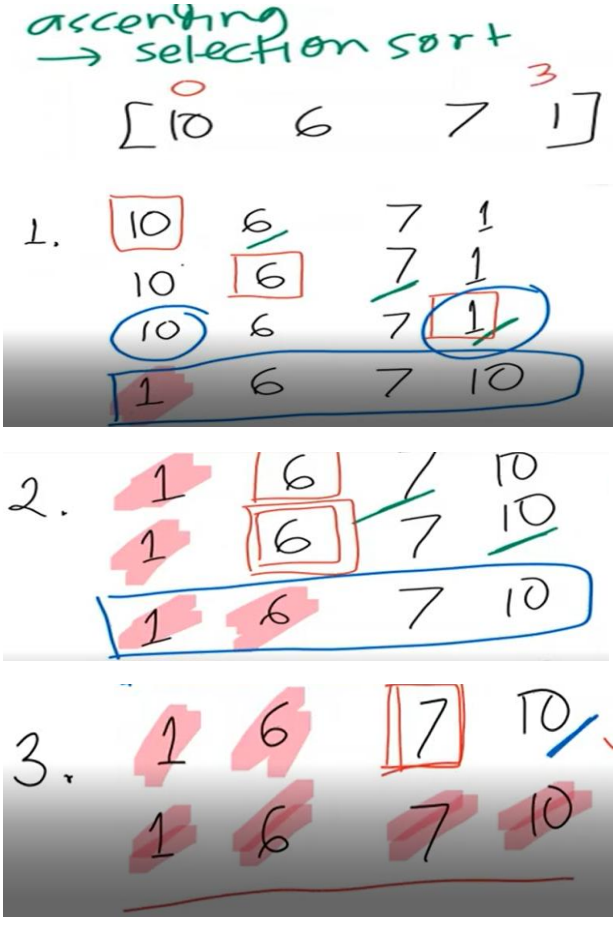


Pada selection sort, proses swap dilakukan stlh smua data dicek.



4 Data

→ 3 outer loop:

→ 1. 3x iterasi

→ 0 → 1

→ 2

→ 3

2. 2x iterasi

→ 1 → 2

→ 3

3. 1x iterasi

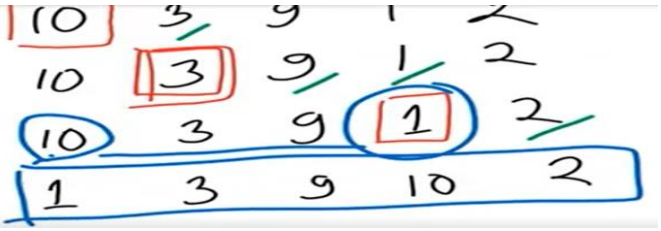
→ 2 → 3

5 data ascending

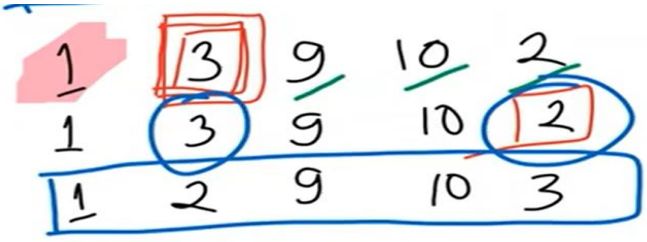
0 1 2 3 4

[10, 3, 9, 1, 2]

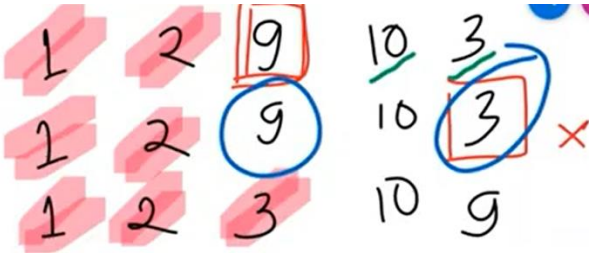
1.

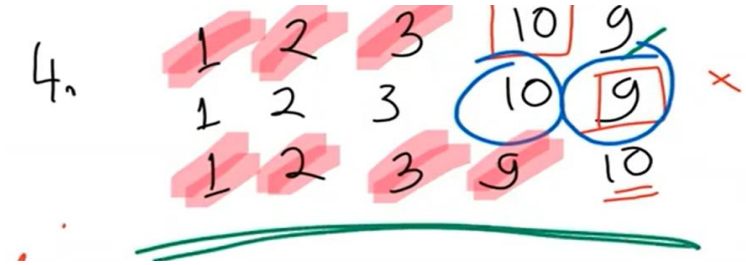


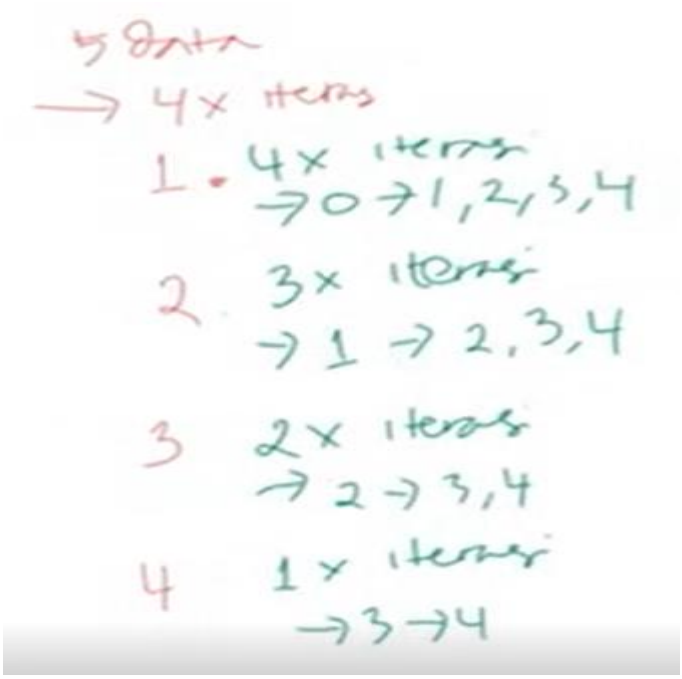
2.



3.







Selection Sort

In [4]:
```
n=4
for outerLoop in range(n-1):
    print (outerLoop)
```

```
0
1
2
```

In [5]:
```
n=4
for outerLoop in range(n-1):
    for j in range(outerLoop+1,n):
        print (outerLoop,j)
```

```
0 1
0 2
0 3
1 2
1 3
2 3
```

In [6]:
```
n=4
data=[10,6,7,1]
for outerLoop in range(len(data)-1):
    minFlag=outerLoop
    for j in range(outerLoop+1,len(data)):
        if data[minFlag]>data[j]:
            minFlag=j
    data[outerLoop],data[minFlag]=data[minFlag],data[outerLoop]
    print (data)
```

```
[1, 6, 7, 10]
[1, 6, 7, 10]
[1, 6, 7, 10]
```

In [7]:
```
n=4
data=[10,3,9,1,2]
for outerLoop in range(len(data)-1):
    minFlag=outerLoop
    for j in range(outerLoop+1,len(data)):
        if data[minFlag]>data[j]:
            minFlag=j
    data[outerLoop],data[minFlag]=data[minFlag],data[outerLoop]
    print (data)
```

```
[1, 3, 9, 10, 2]
[1, 2, 9, 10, 3]
[1, 2, 3, 10, 9]
[1, 2, 3, 9, 10]
```

In [9]:
```
import modifiedSorting
data=[10,3,9,1,2]

print(modifiedSorting.selecSortAscending(data,'max'))
```

```
[10, 3, 9, 1, 2]
iterasi ke- 1 : [2, 3, 9, 1, 10]
iterasi ke- 2 : [2, 3, 1, 9, 10]
iterasi ke- 3 : [2, 1, 3, 9, 10]
iterasi ke- 4 : [1, 2, 3, 9, 10]
[1, 2, 3, 9, 10]
```

In [10]:
```
import modifiedSorting
data=[10,3,9,1,2]

print(modifiedSorting.selecSortAscending(data,'max'))
print(modifiedSorting.selecSortAscending(data,'min'))
```

```
[10, 3, 9, 1, 2]
iterasi ke- 1 : [2, 3, 9, 1, 10]
iterasi ke- 2 : [2, 3, 1, 9, 10]
iterasi ke- 3 : [2, 1, 3, 9, 10]
iterasi ke- 4 : [1, 2, 3, 9, 10]
[1, 2, 3, 9, 10]
[10, 3, 9, 1, 2]
iterasi ke- 1 : [1, 3, 9, 10, 2]
iterasi ke- 2 : [1, 2, 9, 10, 3]
iterasi ke- 3 : [1, 2, 3, 10, 9]
iterasi ke- 4 : [1, 2, 3, 9, 10]
[1, 2, 3, 9, 10]
```

In [11]:
```
print(modifiedSorting.modifiedSelection(data))
```

```
iterasi ke- 1
minimal  [1, 3, 9, 10, 2]
maksimal  [1, 3, 9, 2, 10]
iterasi ke- 2
minimal  [1, 2, 9, 3, 10]
maksimal  [1, 2, 3, 9, 10]
[1, 2, 3, 9, 10]
```

In [12]:
```
data=[100,12,3,4,5,99,12,30,45,6,7,2,3,4,5,1]
print(modifiedSorting.modifiedSelection(data))
```

```
iterasi ke- 1
minimal  [1, 12, 3, 4, 5, 99, 12, 30, 45, 6, 7, 2, 3, 4, 5, 100]
maksimal  [1, 12, 3, 4, 5, 99, 12, 30, 45, 6, 7, 2, 3, 4, 5, 100]
iterasi ke- 2
minimal  [1, 2, 3, 4, 5, 99, 12, 30, 45, 6, 7, 12, 3, 4, 5, 100]
maksimal  [1, 2, 3, 4, 5, 5, 12, 30, 45, 6, 7, 12, 3, 4, 99, 100]
iterasi ke- 3
minimal  [1, 2, 3, 4, 5, 5, 12, 30, 45, 6, 7, 12, 3, 4, 99, 100]
maksimal  [1, 2, 3, 4, 5, 5, 12, 30, 4, 6, 7, 12, 3, 45, 99, 100]
iterasi ke- 4
minimal  [1, 2, 3, 3, 5, 5, 12, 30, 4, 6, 7, 12, 4, 45, 99, 100]
maksimal  [1, 2, 3, 3, 5, 5, 12, 4, 4, 6, 7, 12, 30, 45, 99, 100]
iterasi ke- 5
minimal  [1, 2, 3, 3, 4, 5, 12, 5, 4, 6, 7, 12, 30, 45, 99, 100]
maksimal  [1, 2, 3, 3, 4, 5, 12, 5, 4, 6, 7, 12, 30, 45, 99, 100]
iterasi ke- 6
minimal  [1, 2, 3, 3, 4, 4, 12, 5, 5, 6, 7, 12, 30, 45, 99, 100]
maksimal  [1, 2, 3, 3, 4, 4, 7, 5, 5, 6, 12, 12, 30, 45, 99, 100]
iterasi ke- 7
minimal  [1, 2, 3, 3, 4, 4, 5, 7, 5, 6, 12, 12, 30, 45, 99, 100]
maksimal  [1, 2, 3, 3, 4, 4, 5, 6, 5, 7, 12, 12, 30, 45, 99, 100]
iterasi ke- 8
minimal  [1, 2, 3, 3, 4, 4, 5, 5, 6, 7, 12, 12, 30, 45, 99, 100]
maksimal  [1, 2, 3, 3, 4, 4, 5, 5, 6, 7, 12, 12, 30, 45, 99, 100]
[1, 2, 3, 3, 4, 4, 5, 5, 6, 7, 12, 12, 30, 45, 99, 100]
```

Beberapa modifikasi yang bisa dilakukan utk Selection Sort:

1. Ascending, cari nilai maks dan letakkan di indeks-indeks akhir
2. Descending, cari nilai min dan letakkan di indeks-indeks akhir
3. Descending, cari nilai maks dan letakkan di indeks-indeks awal
4. Ascending, sekali iterasi luar, lakukan dua kali proses yaitu cari nilai maks dan letakkan di indeks akhir serta cari nilai indeks min & letakkan di indeks awal

Latihan Insertion Sort:

1. Buat code descending
2. Ascending tetapi proses pengurutan dimulai dari indeks-indeks terakhir, bukan pertama
3. Descending tetapi proses pengurutan dimulai dari indeks-indeks terakhir, bukan pertama