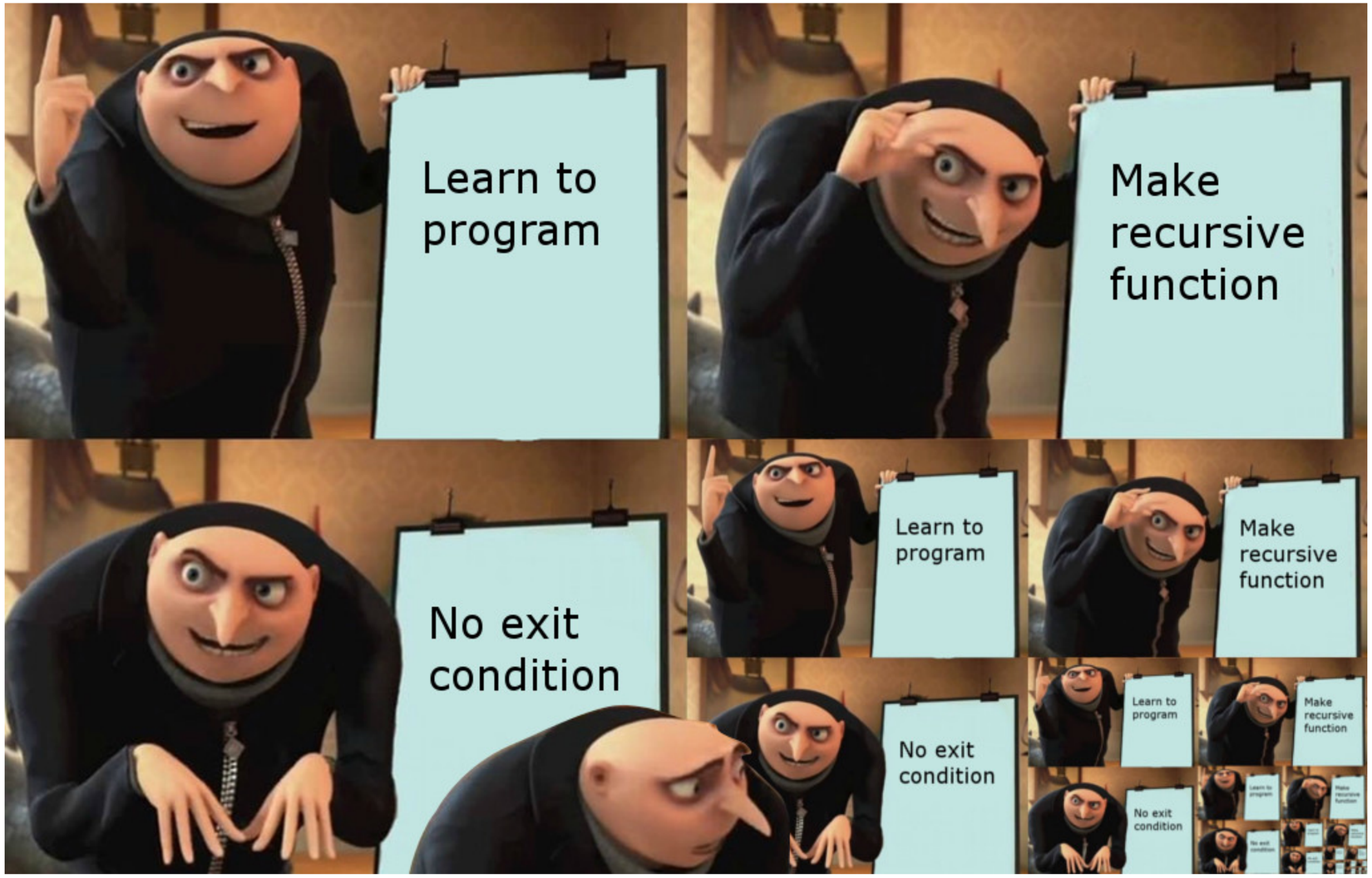
Web Development with Python Lesson 5

Ing. Patrik Dendis

STEP IT ACADEMY BRATISLAVA


Learn to program
Make recursive function
No exit condition
Learn to program
Make recursive function
No exit condition
Learn to program
Make recursive function
No exit condition

OBSAH PREZENTÁCIE

- Opakovanie
- Sorting
- Bubble Sort
- Porovnanie algoritmov
- Search
- Linear search
- Binary search

OPAKOVANIE

- Ako pridáte do projektu matematické funkcie?
- Aké kľúčové slovo maju funkcie?
- Ako sa označuje parameter kde nevieme koľko parametrov bude funkcia mať?
- Čo je rekurzia?
- Čo sú lambda funkcie?
- Čo robí map a filter funkcia?
- Načo máme dekorátory?

TRIEDENIE - SORTING

- triedenie je proces zoradenia prvkov v zozname alebo poli do určitého poradia (napríklad vzostupné alebo zostupné).

- dôležitosti triedenia v oblastiach ako databázy, vyhľadávanie, optimalizácia a ďalšie

PYTHON SORTING

- the **sort()** method sorts the list ascending by default

main.py > ...

```
1  cisla = [23, 2, 1, 5, 3, 15, 7]
2  cisla.sort()
3  print(cisla)
```

Run

```
[1, 2, 3, 5, 7, 15, 23]
```

main.py

main.py > ...

```
1  cisla = [23, 2, 1, 5, 3, 15, 7]
2  cisla.sort(reverse=True)
3  print(cisla)
```

Console Shell

Run

```
[23, 15, 7, 5, 3, 2, 1]
```

PYTHON SORTING

main.py

```
1  mena = ["Patrik", "Milan", "Peter", "Lukas", "Simon"]
2  mena.sort()
3  print(mena)
```

Console | Shell

Run

```
['Lukas', 'Milan', 'Patrik', 'Peter', 'Simon']
```

main.py

```
1  def zorad_podla_dlzky(list):
2      return len(list)
3
4  mena = ["Patrik", "Milan", "Peter", "Lukas", "Ivan",
   "Kristian"]
5  mena.sort(key=zorad_podla_dlzky)
6  print(mena)
```

Run

```
['Ivan', 'Milan', 'Peter', 'Lukas', 'Patrik', 'Kristian']
```

SORT vs SORTED

main.py > ...

```
1  nums = [2, 3, 1, 5, 6, 4, 0]
2
3  print(sorted(nums))
4  print(nums)
5
6  print(nums.sort())
7  print(nums)
```

Run

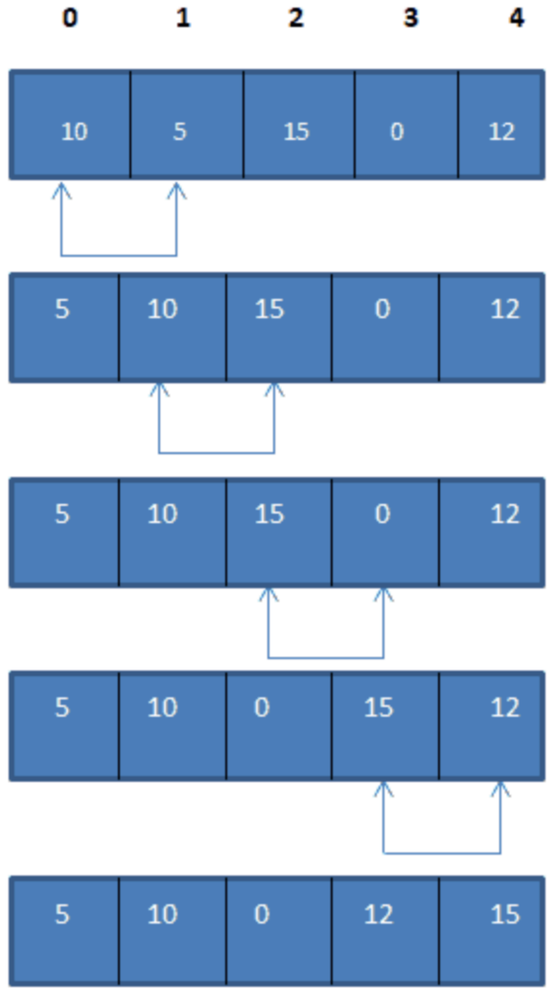
```
[0, 1, 2, 3, 4, 5, 6]
[2, 3, 1, 5, 6, 4, 0]
None
[0, 1, 2, 3, 4, 5, 6]
```

BUBBLE SORT

https://www.youtube.com/watch?v=lv3vgjM8Pv4

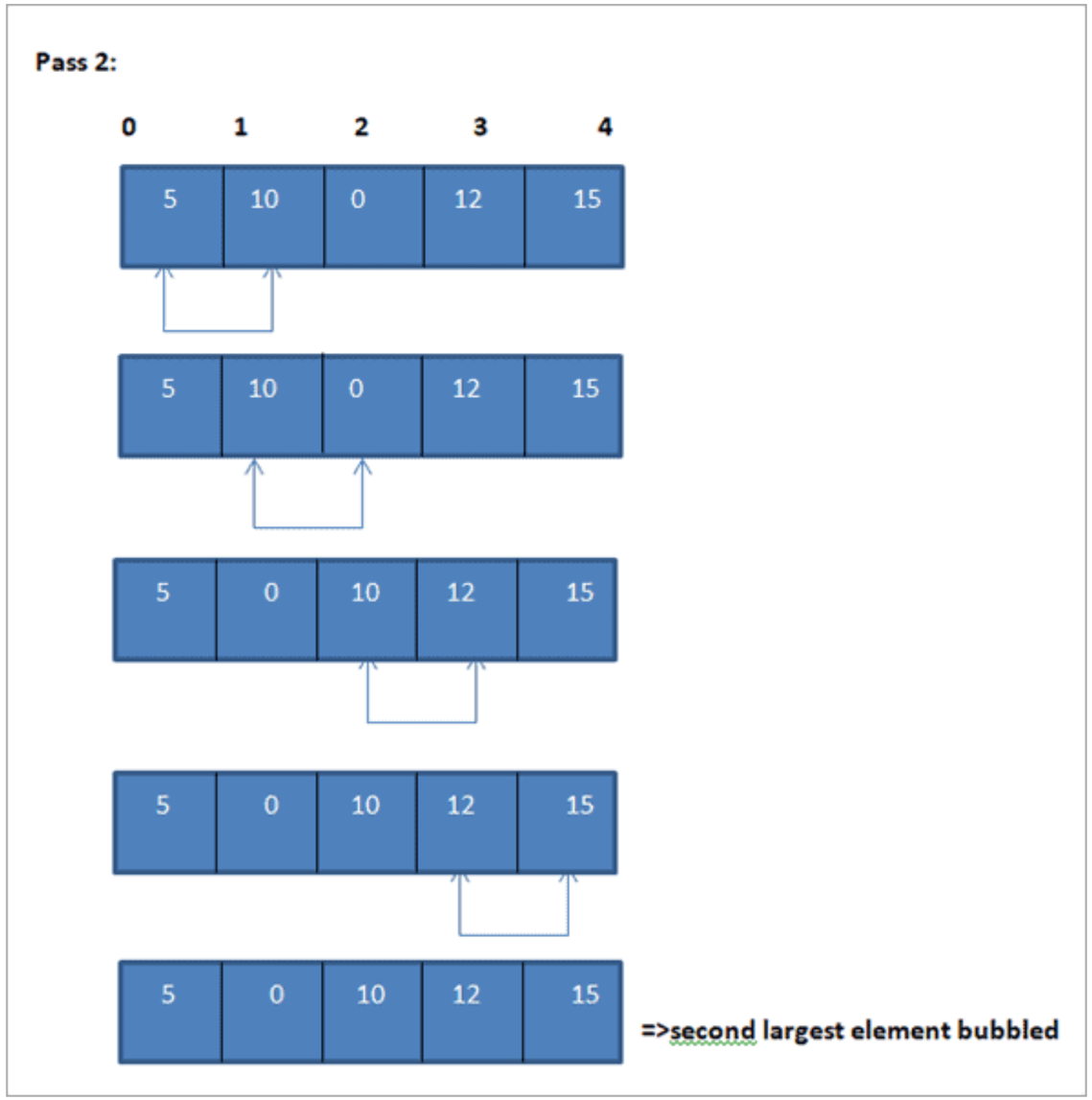
BUBBLE SORT

Pass 1:



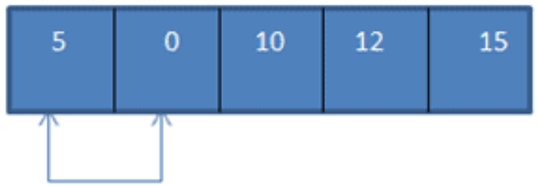
=>the largest element bubbled up

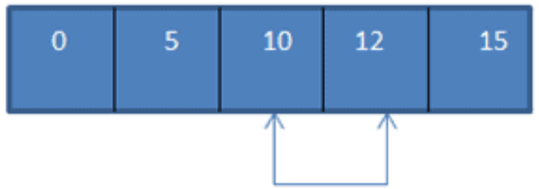
BUBBLE SORT

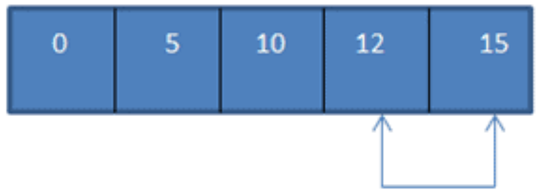

Pass 2:
0 1 2 3 4
5 10 0 12 15
5 10 0 12 15
5 0 10 12 15
5 0 10 12 15
5 0 10 12 15
=>second largest element bubbled

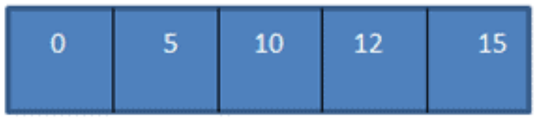
BUBBLE SORT

Pass 3:









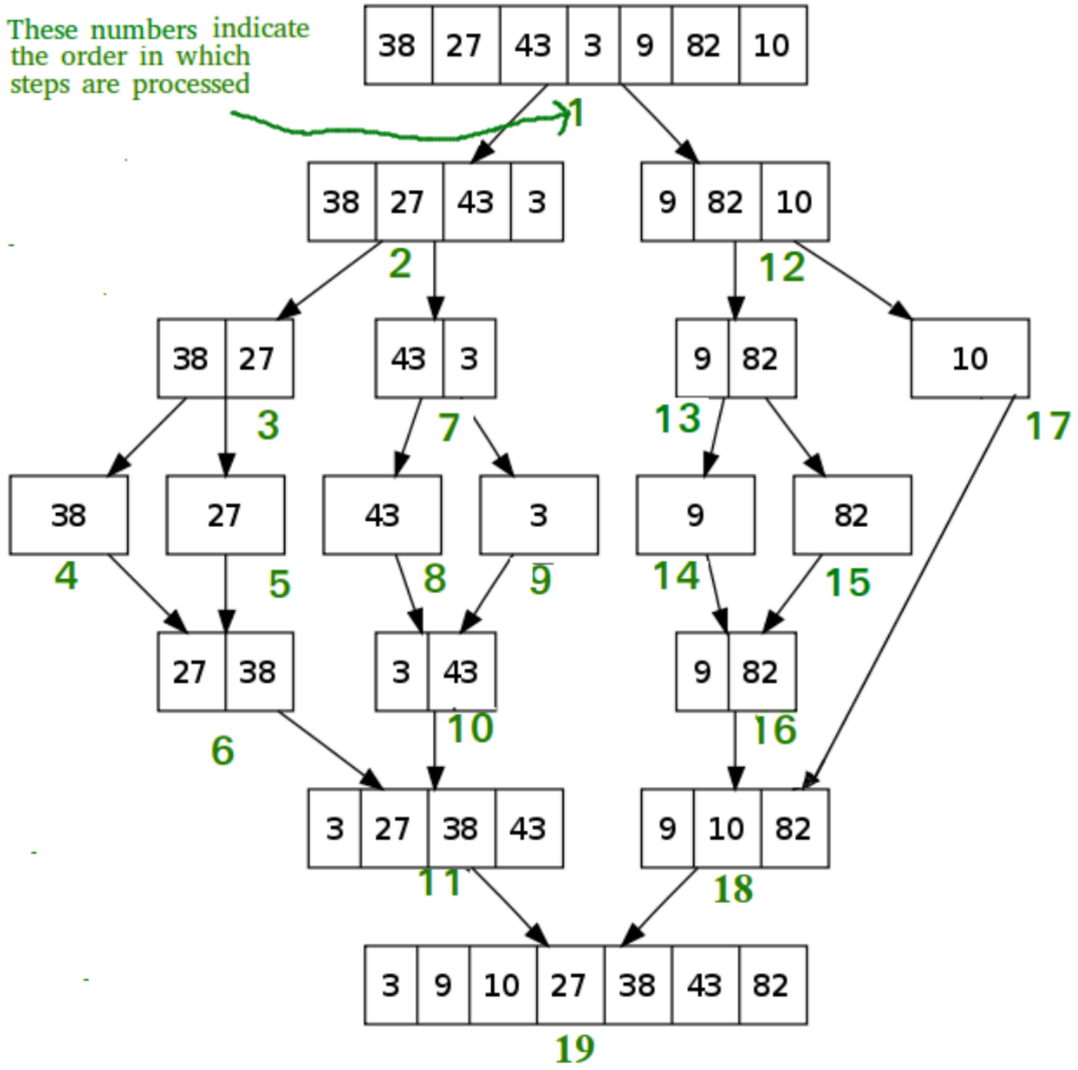
=>third element bubbled up, list sorted

BUBBLE SORT

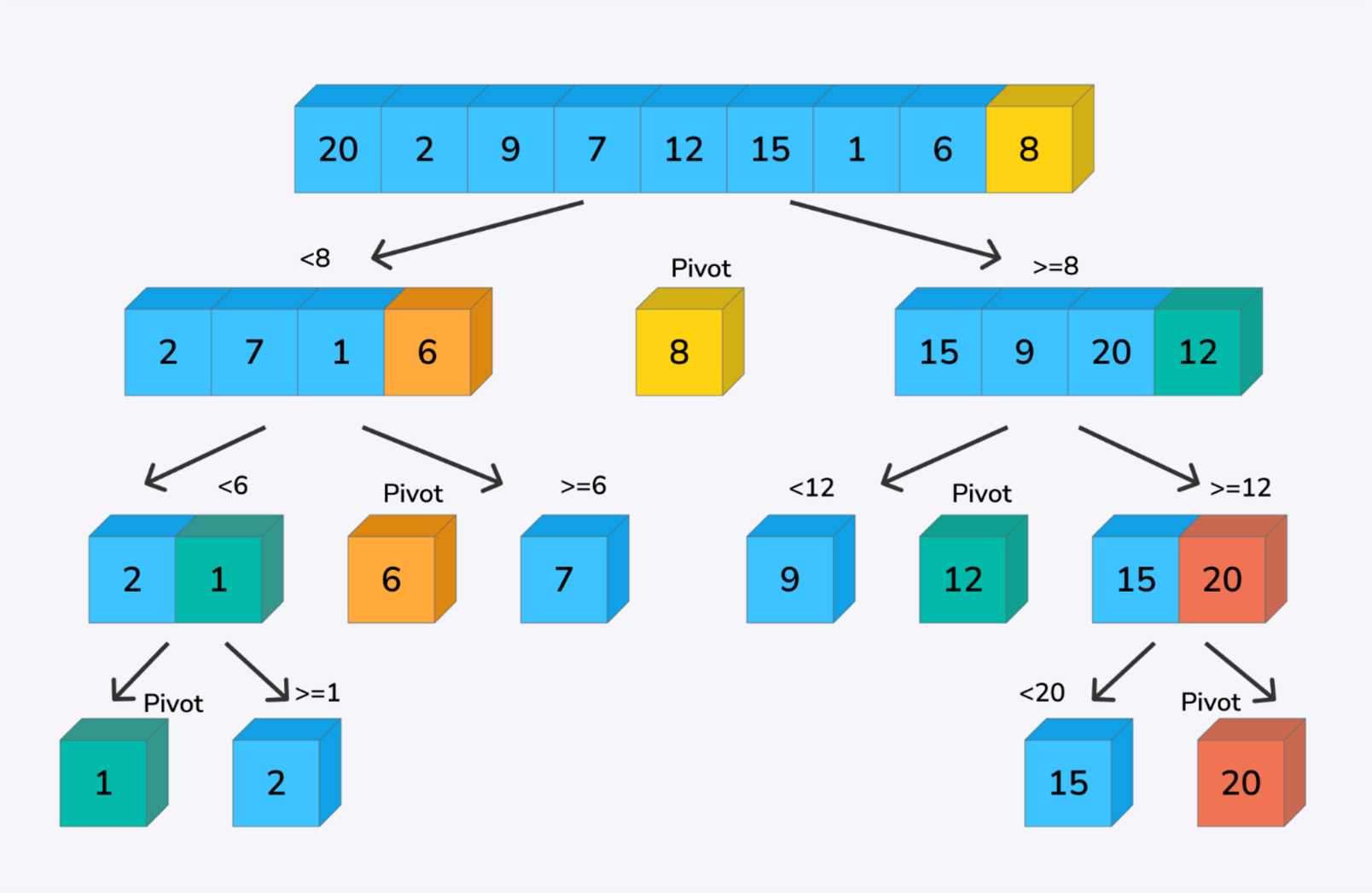
Algorithm 1: Bubble sort

```
Data: Input array A[]
Result: Sorted A[]
int i, j, k;
N = length(A);
for j = 1 to N do
    for i = 0 to N-1 do
        if A[i] > A[i+1] then
            temp = A[i];
            A[i] = A[i+1];
            A[i+1] = temp;
        end
    end
end
```

MERGE SORT


These numbers indicate the order in which steps are processed
38 27 43 3 9 82 10
1
38 27 43 3
9 82 10
2
12
38 27
43 3
9 82
10
3
7
13
17
38
27
43
3
9
82
4
5
8
9
14
15
27 38
3 43
9 82
10
16
6
3 27 38 43
9 10 82
11
18
3 9 10 27 38 43 82
19

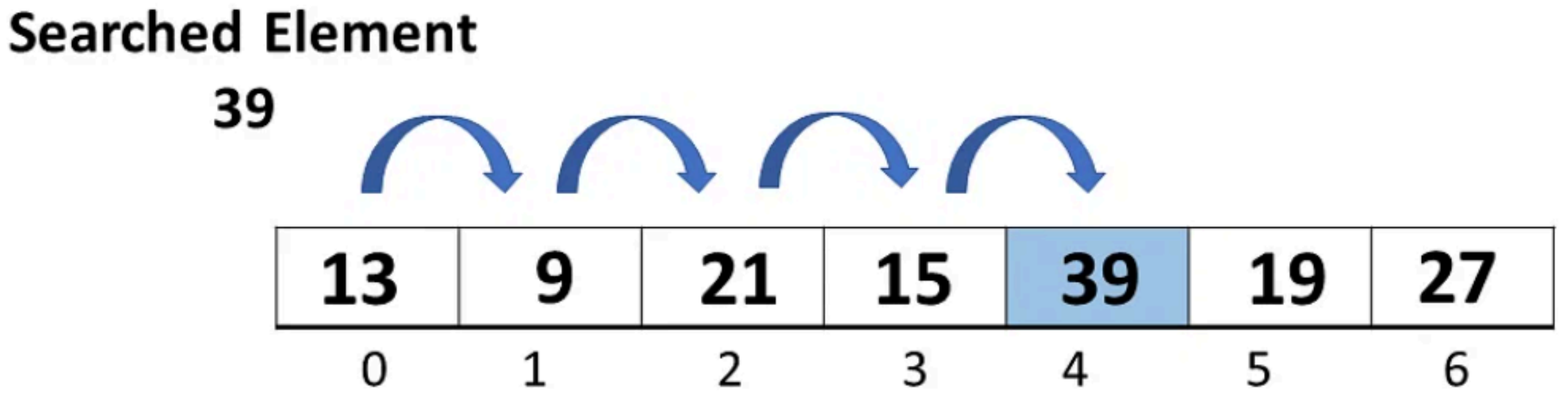
QUICK SORT


20 2 9 7 12 15 1 6 8
<8
Pivot
>=8
2 7 1 6
8
15 9 20 12
<6
Pivot
>=6
<12
Pivot
>=12
2 1
6
7
9
12
15 20
Pivot
>=1
<20
Pivot
1
2
15
20

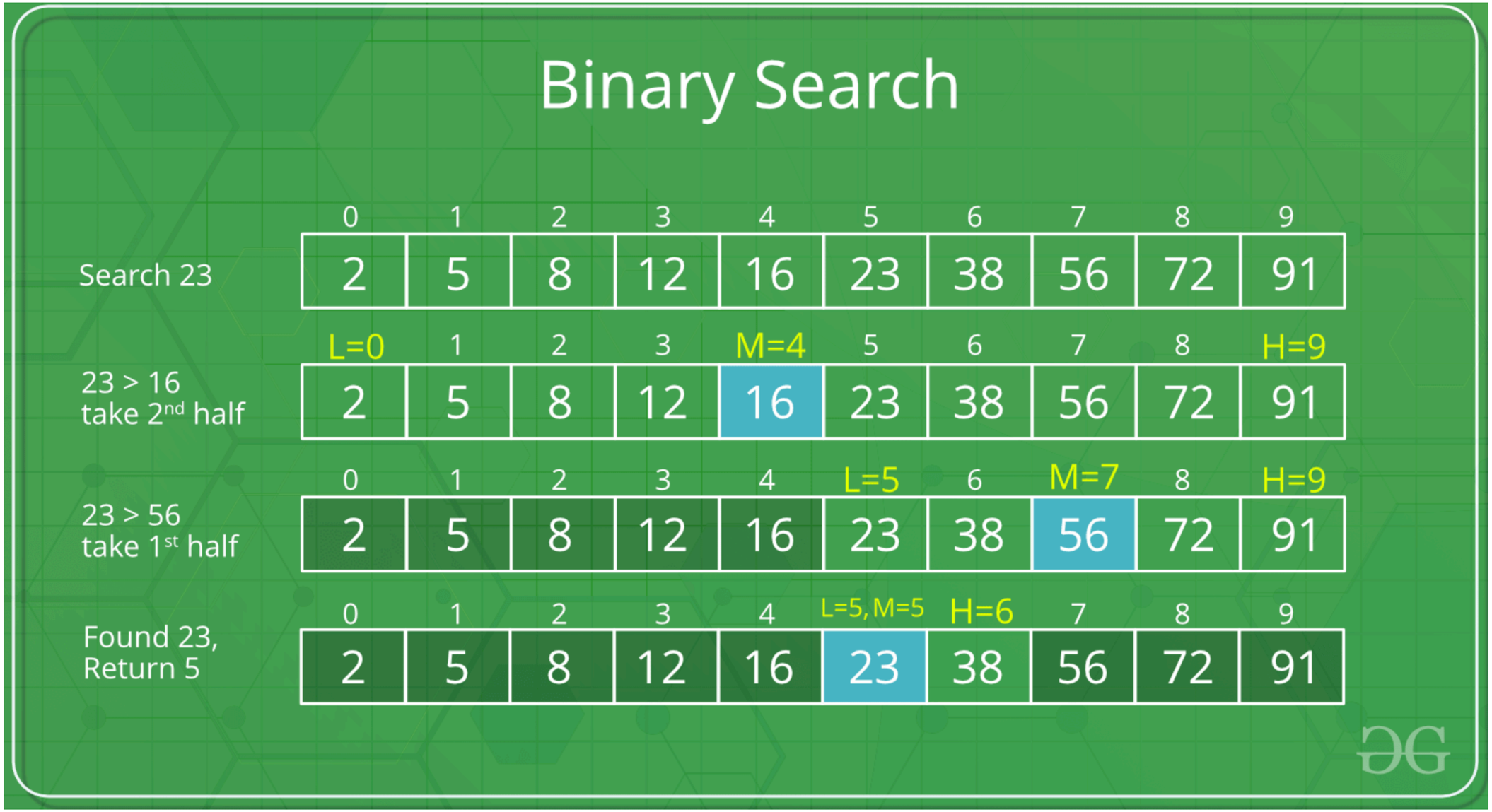
POROVNANIE

Sorting Algorithms	Time Complexity			Space Complexity
	Best Case	Average Case	Worst Case	Worst Case
Bubble Sort	Ω(N)	Θ(N^2)	O(N^2)	O(1)
Selection Sort	Ω(N^2)	Θ(N^2)	O(N^2)	O(1)
Insertion Sort	Ω(N)	Θ(N^2)	O(N^2)	O(1)
Quick Sort	Ω(N log N)	Θ(N log N)	O(N^2)	O(N)
Merge Sort	Ω(N log N)	Θ(N log N)	O(N log N)	O(N)
Heap Sort	Ω(N log N)	Θ(N log N)	O(N log N)	O(1)

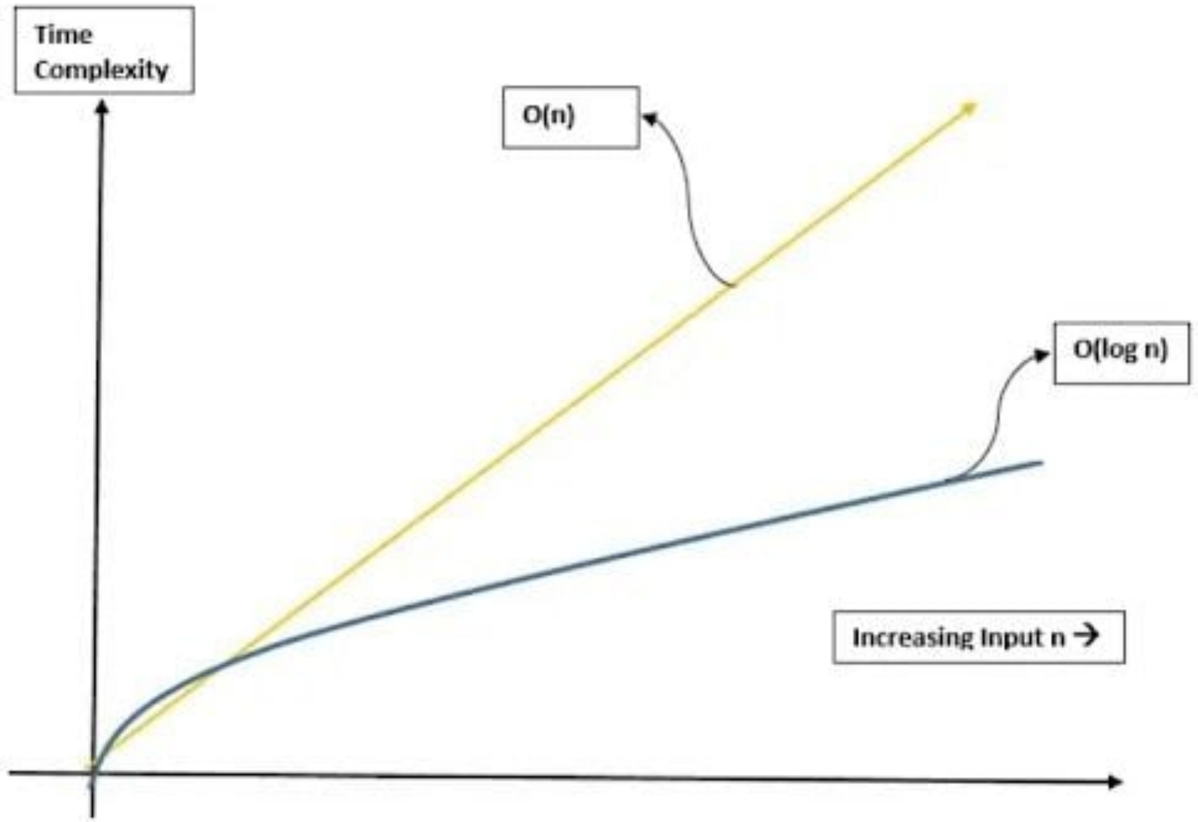
LINEÁRNE HĽADANIE



BINÁRNE HĽADANIE



POROVNANIE HĽADANIE



ZADANIE

- Vytvorte 4 zoznamy kde bude minimálne 5 celý čísel.
- Skombinujte ich do piateho zoznamu.
- Napíšte funkciu ktorá zoradí skombinovaný zoznam podla parametra vzostupne alebo zostupne.
- Napíšte funkciu nájde hodnotu zadanú používateľom pomocou lineárneho vyhľadávania.

ĎAKUJEM ZA POZORNOSŤ