GTU Department of Computer Engineering
CSE 222/505 - Spring 2020
#Question 1

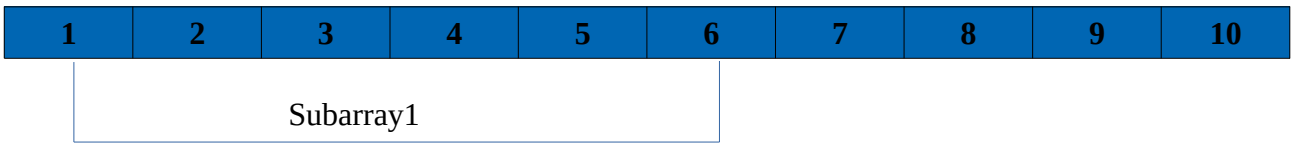
Mert Can BEŞİRLİ
1801042663

#Shell Sort#

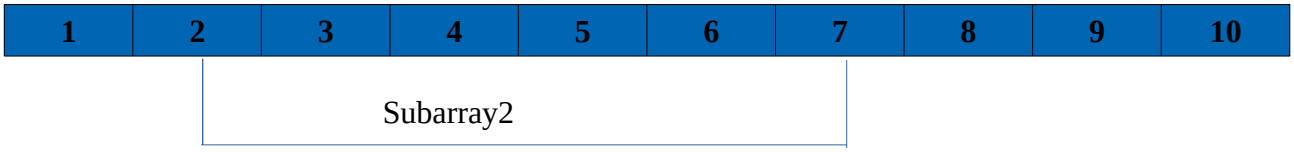
*A={1,2,3,4,5,6,7,8,9,10}

Gap value	5

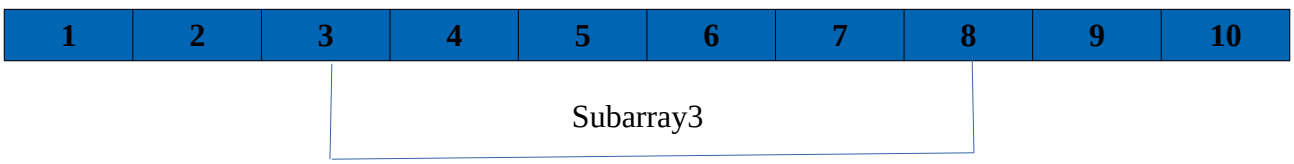
Sorting Subarray1



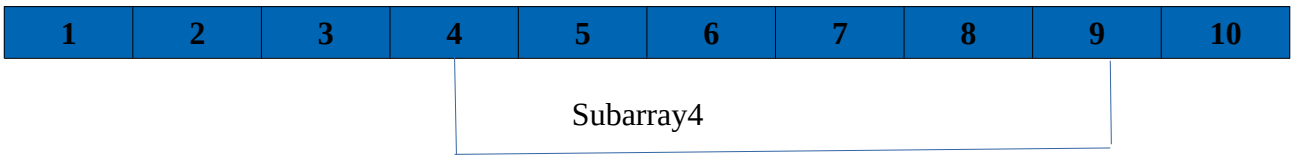
Sorting Subarray2



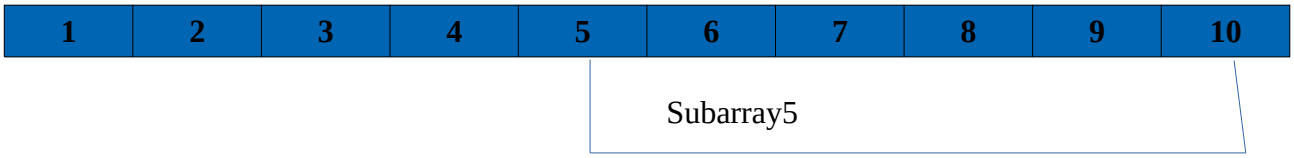
Sorting Subarray3



Sorting Subarray4



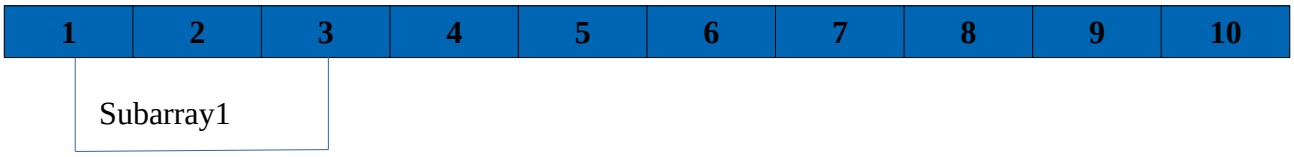
Sorting Subarray5



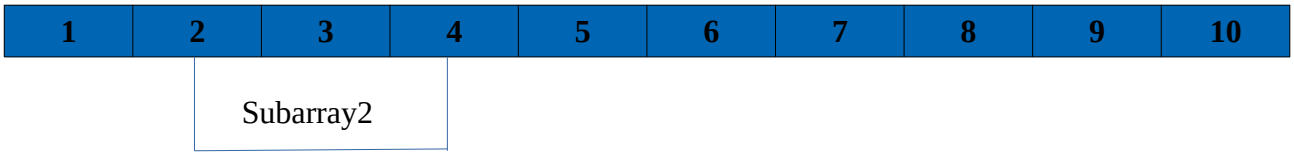
Sort on smaller gap next value

Gap value	**2**

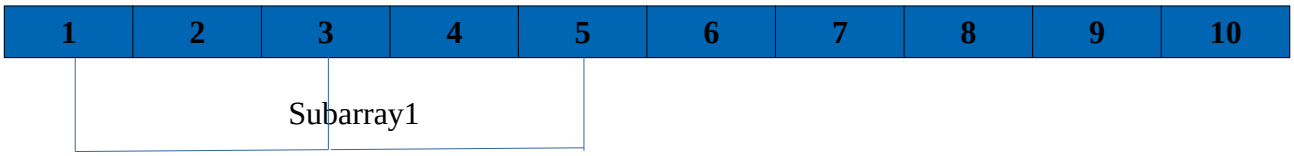
Sorting Subarray1



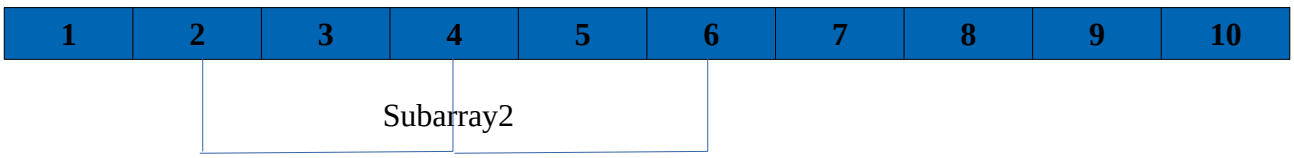
Sorting Subarray2



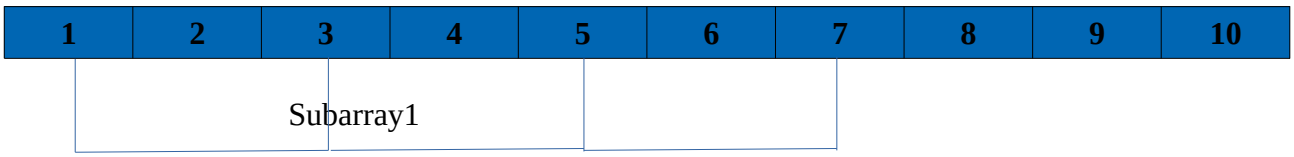
Sorting Subarray1



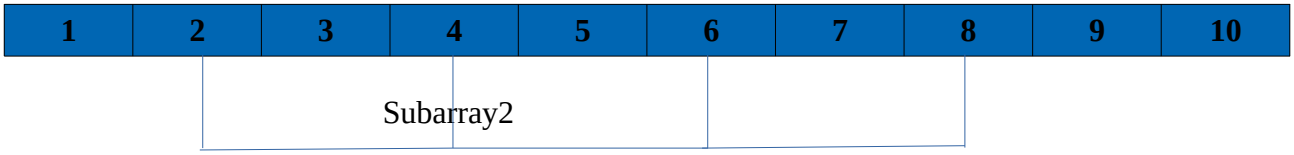
Sorting Subarray2



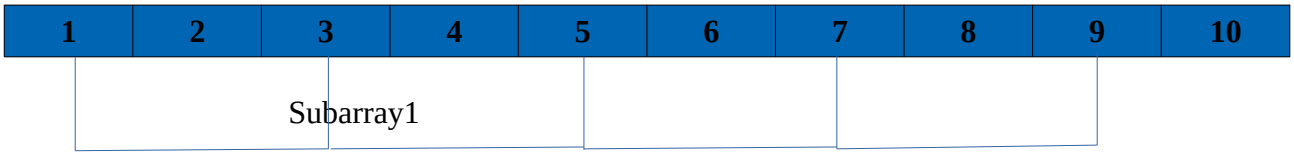
Sorting Subarray1



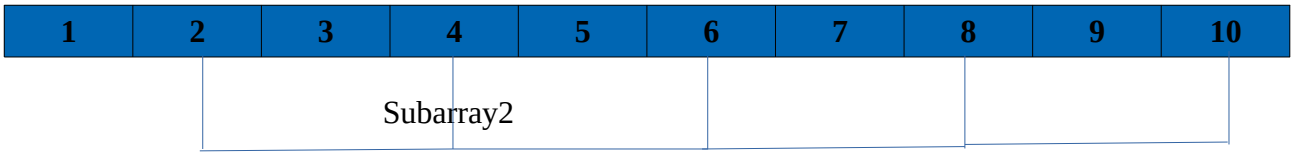
Sorting Subarray2



Sorting Subarray1

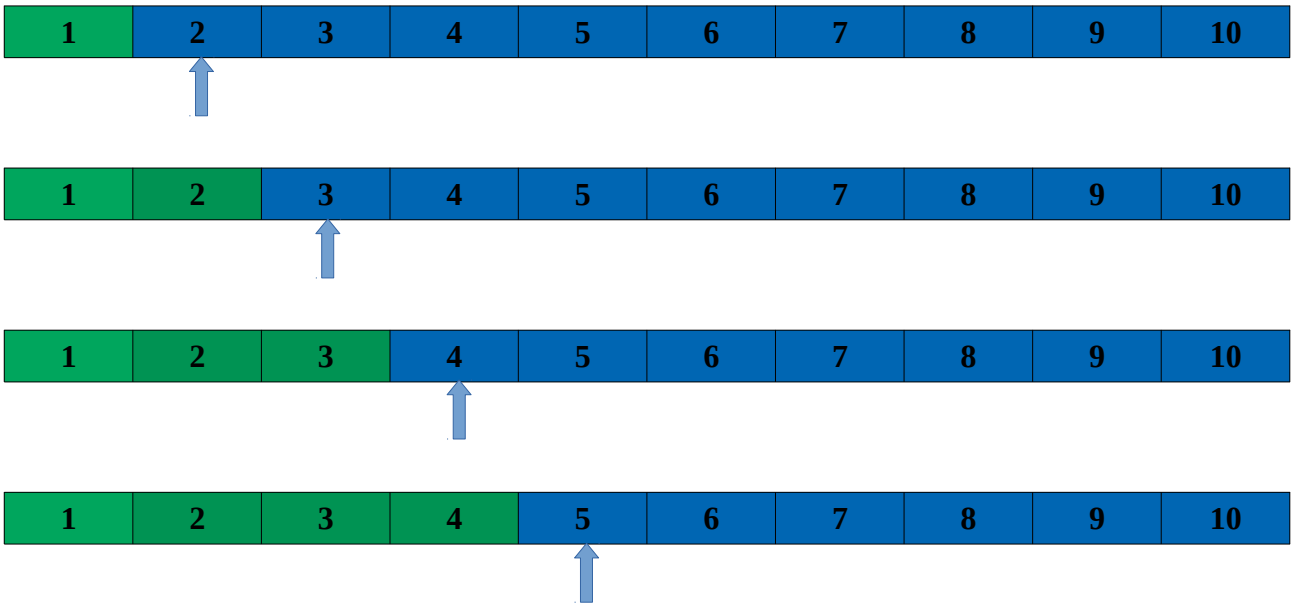


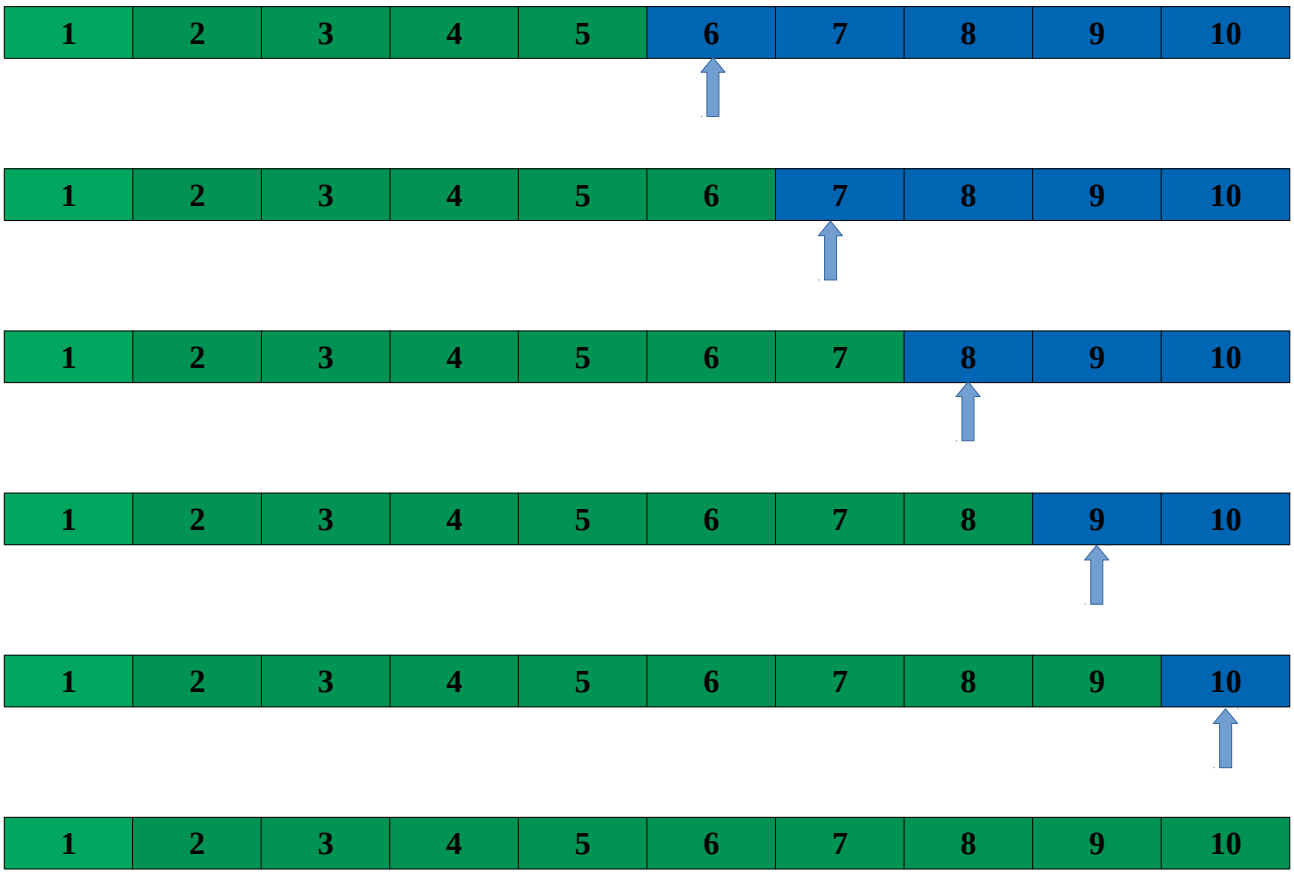
Sorting Subarray2



Sort on gap value of 1(a regular insertion sort)

Gap value	**1**

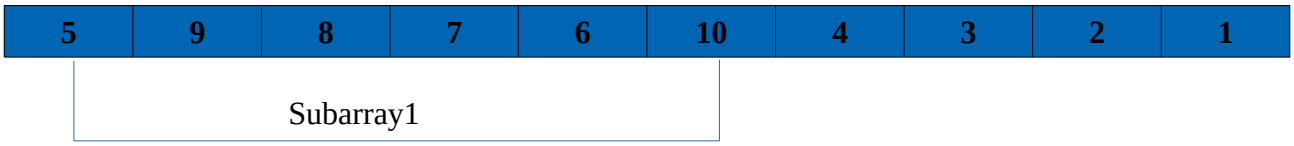




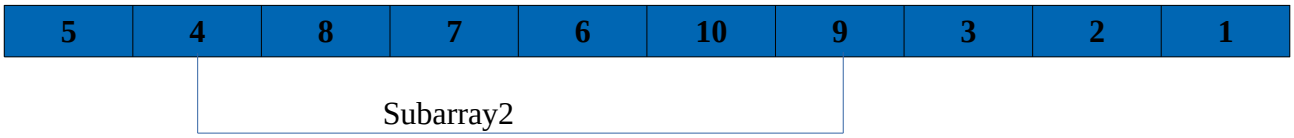
*B={10,9,8,7,6,5,4,3,2,1}

Gap value	**5**

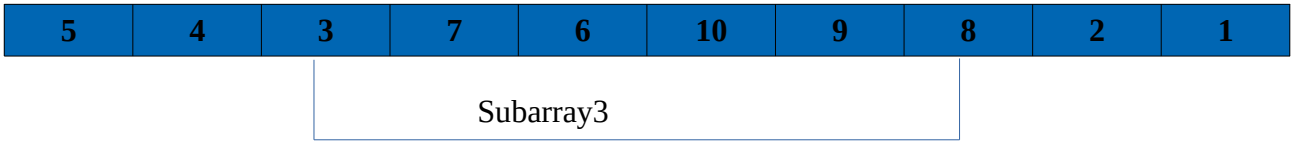
Sorting Subarray1



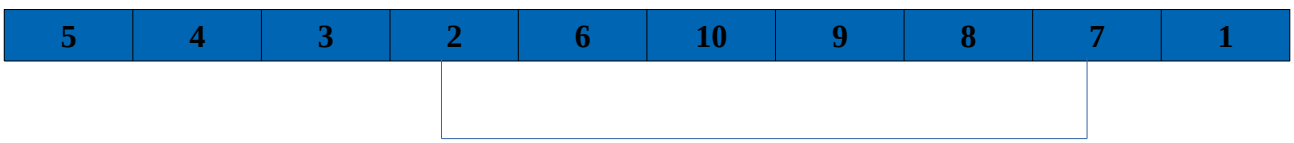
Sorting Subarray2



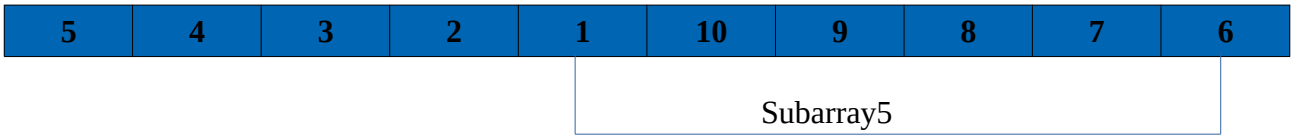
Sorting Subarray3



Sorting Subarray4



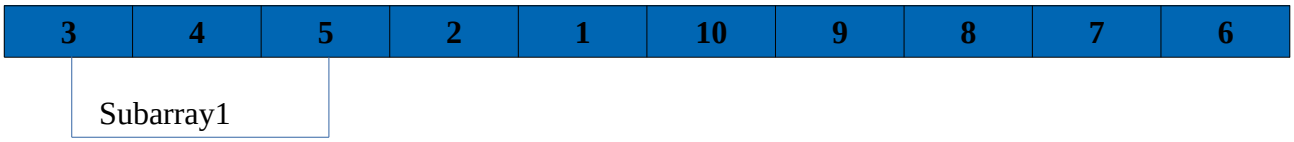
Sorting Subarray5



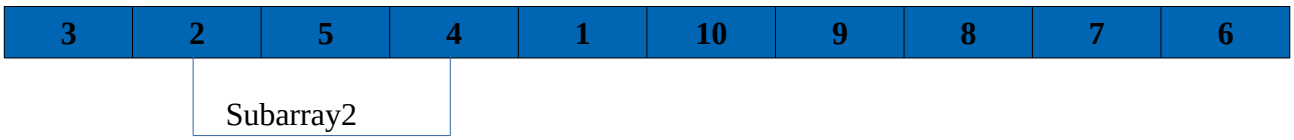
Sort on smaller gap next value

Gap value	**2**

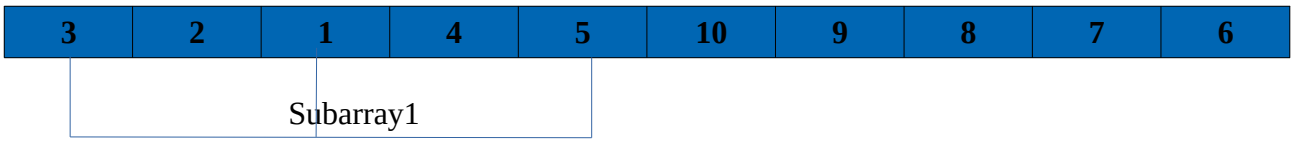
Sorting Subarray1

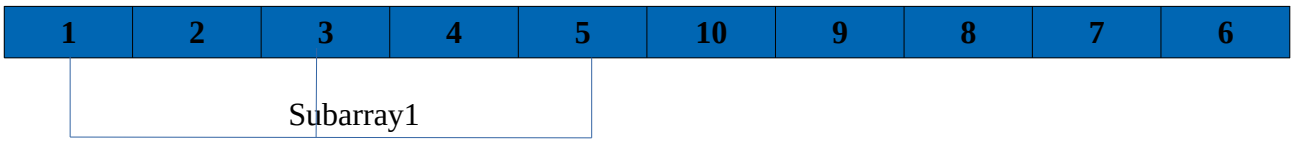


Sorting Subarray2

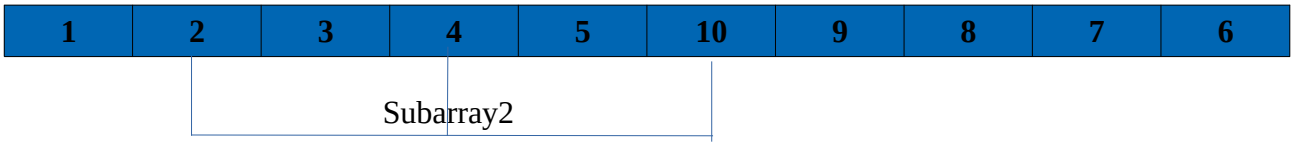


Sorting Subarray1

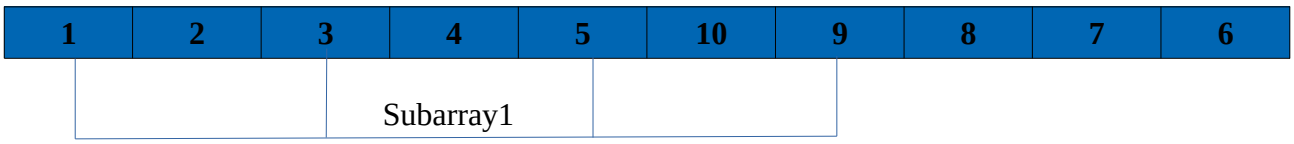




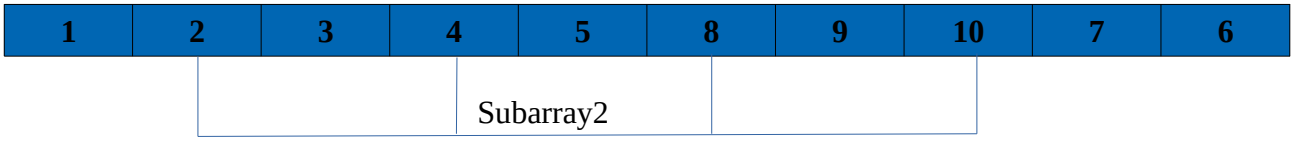
Sorting Subarray2



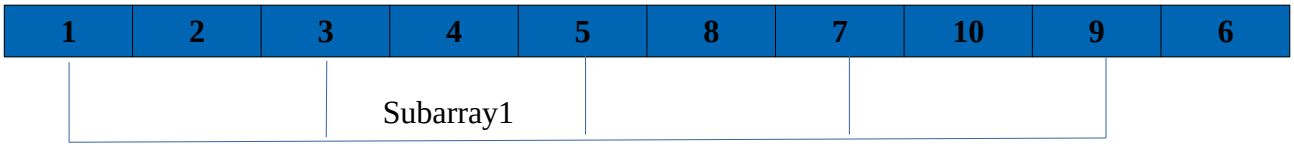
Sorting Subarray1



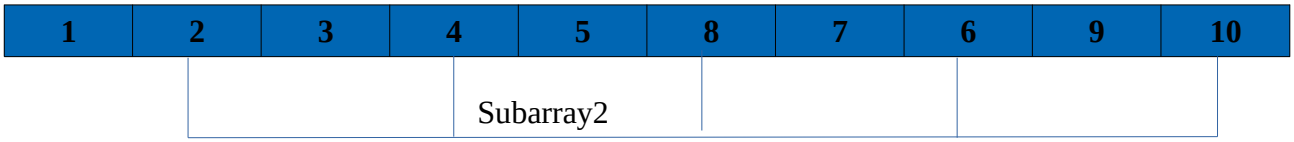
Sorting Subarray2

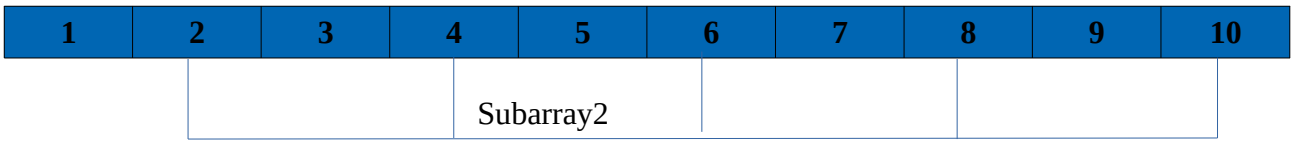


Sorting Subarray1



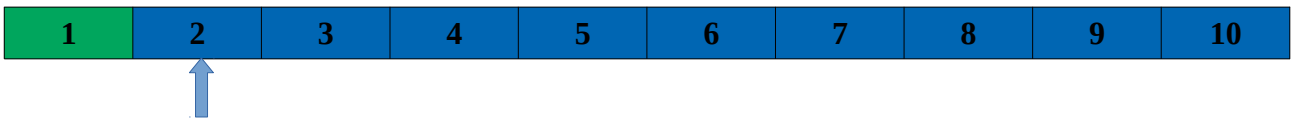
Sorting Subarray2

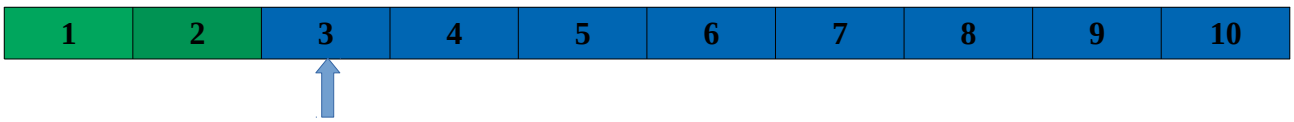


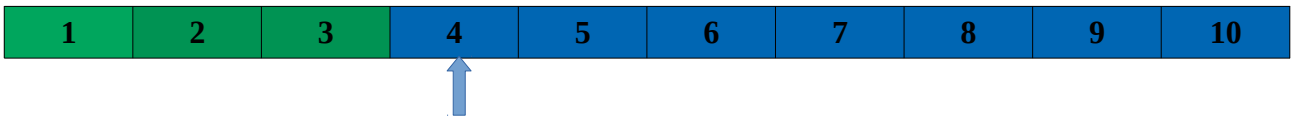


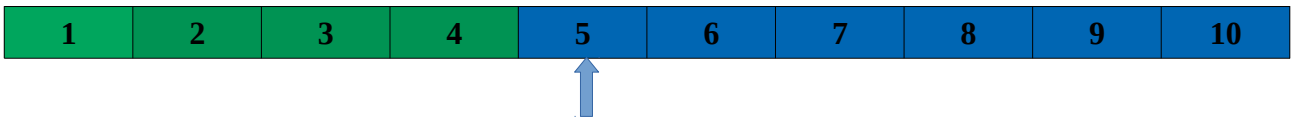
Sort on gap value of 1(a regular insertion sort)

Gap value	**1**





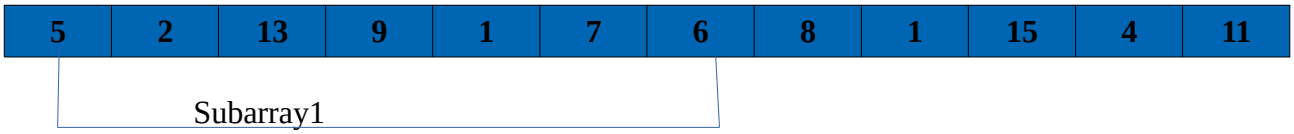




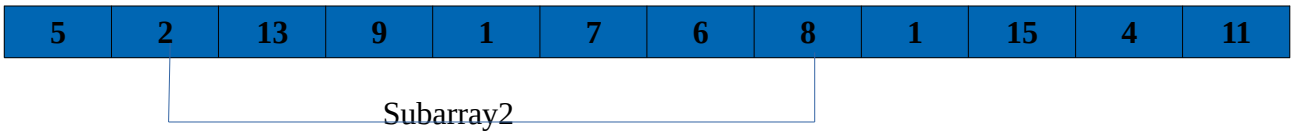
*C={5,2,13,9,1,7,6,8,1,15,4,11}

Gap value	6

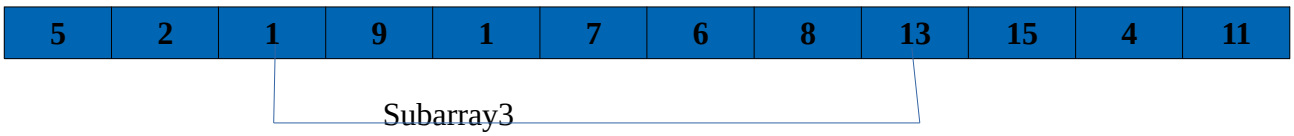
Sorting Subarray1



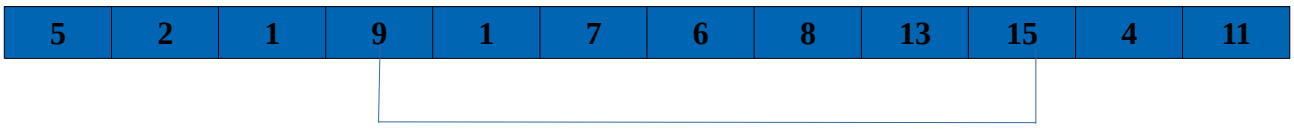
Sorting Subarray2



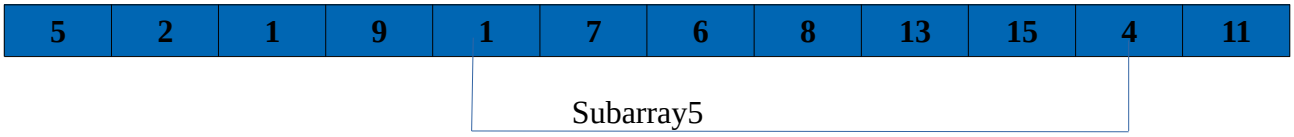
Sorting Subarray3



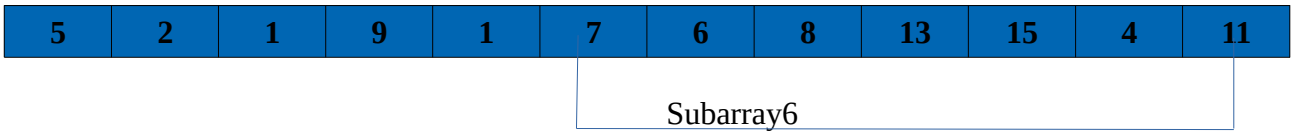
Sorting Subarray4



Sorting Subarray5



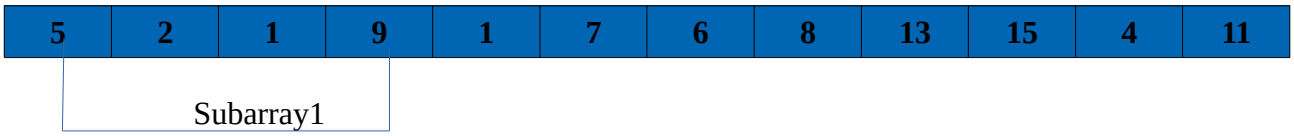
Sorting Subarray6



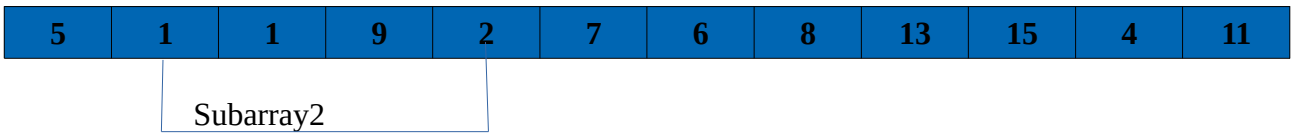
Sort on smaller gap next value

Gap value	**3**

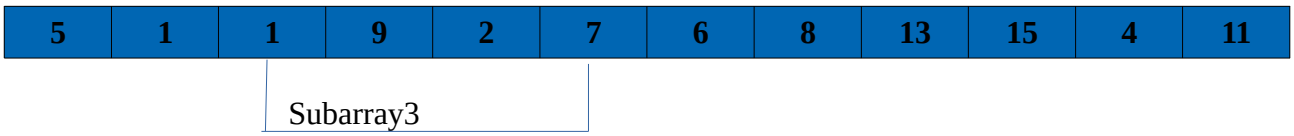
Sorting Subarray1



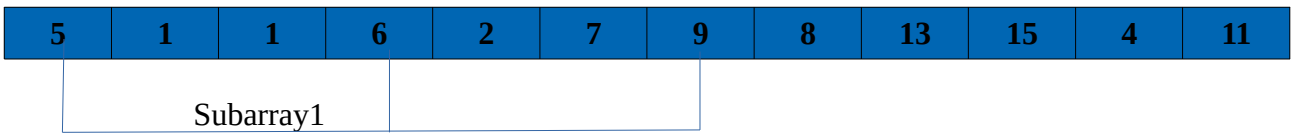
Sorting Subarray2



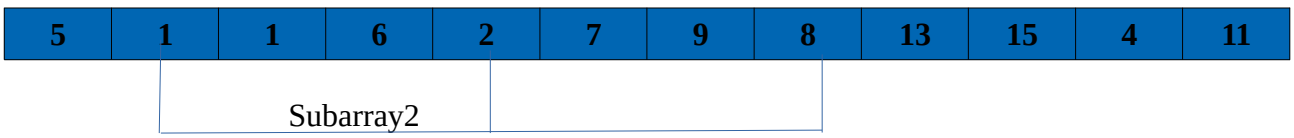
Sorting Subarray3



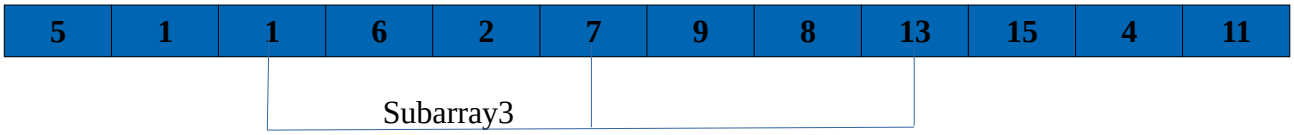
Sorting Subarray1



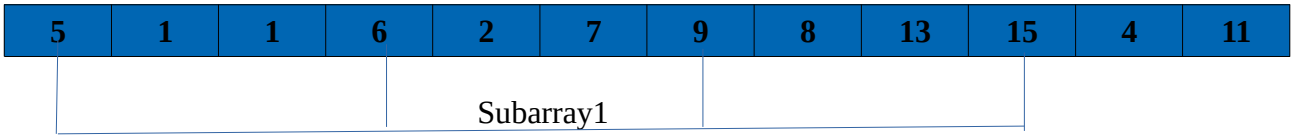
Sorting Subarray2



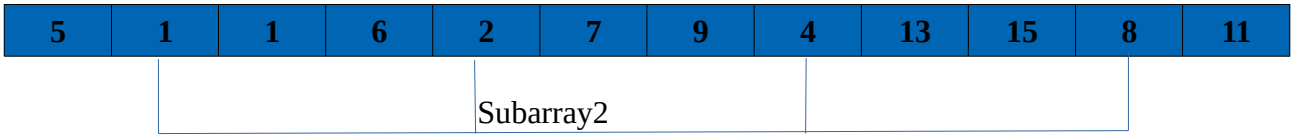
Sorting Subarray3



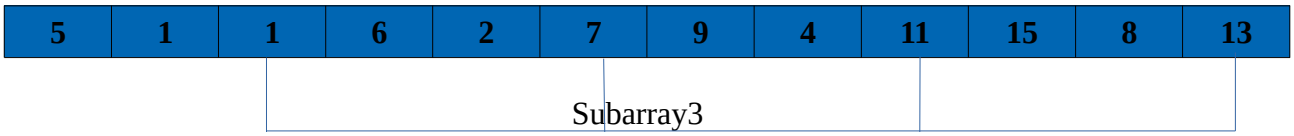
Sorting Subarray1



Sorting Subarray2

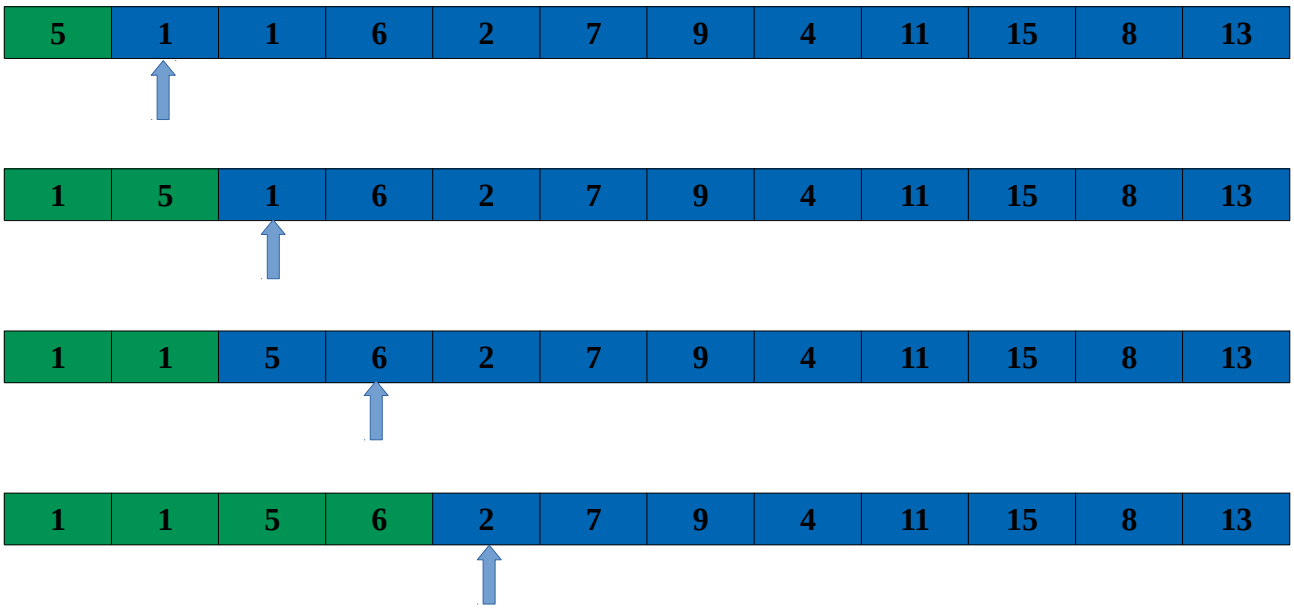


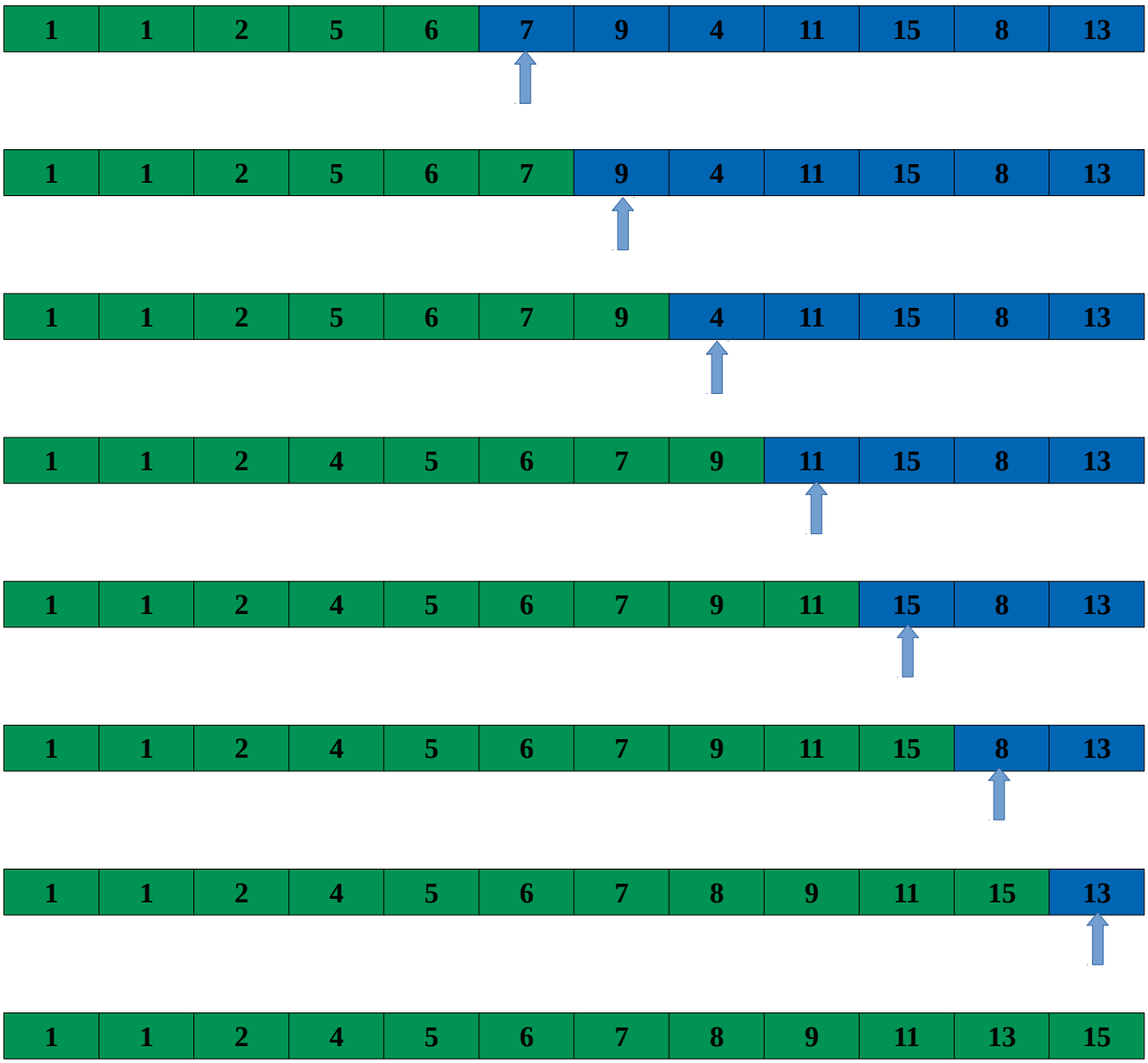
Sorting Subarray3



Sort on gap value of 1(a regular insertion sort)

Gap value	**1**

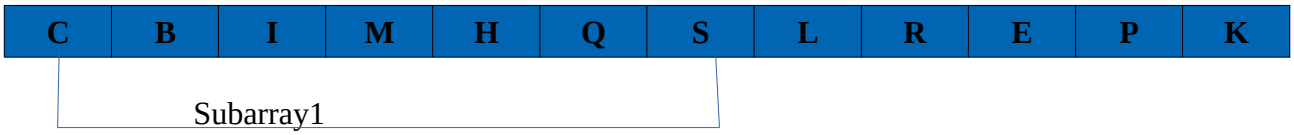




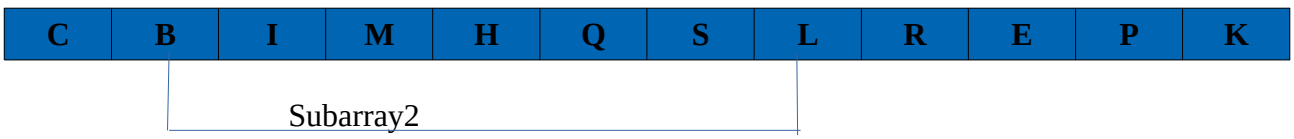
*D={‘S’,’B’,’I’,’M’,’H’,’Q’,’C’,’L’,’R’,’E’,’P’,’K’}

Gap value	**6**

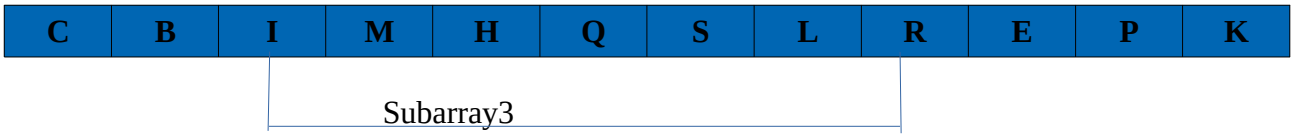
Sorting Subarray1



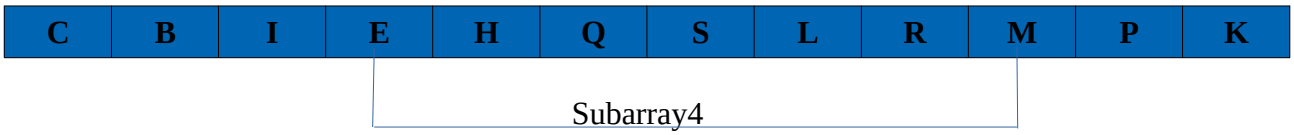
Sorting Subarray2



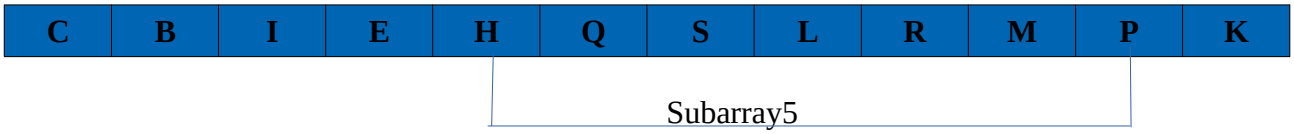
Sorting Subarray3



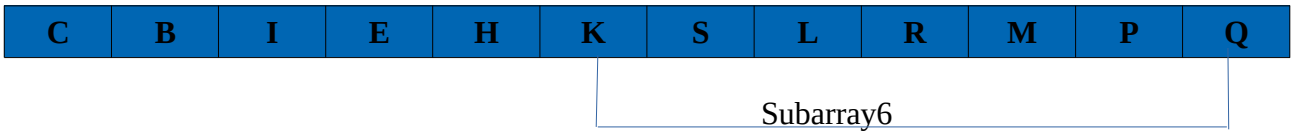
Sorting Subarray4



Sorting Subarray5



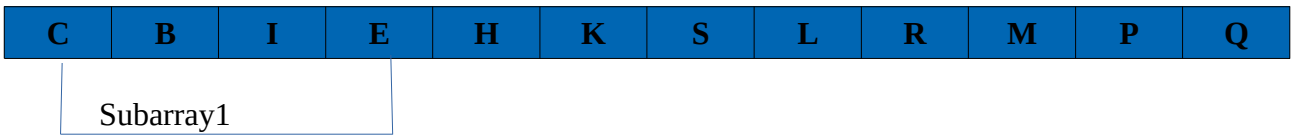
Sorting Subarray6



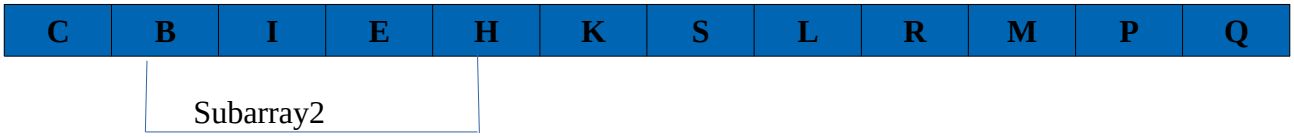
Sort on smaller gap next value

Gap value	**3**

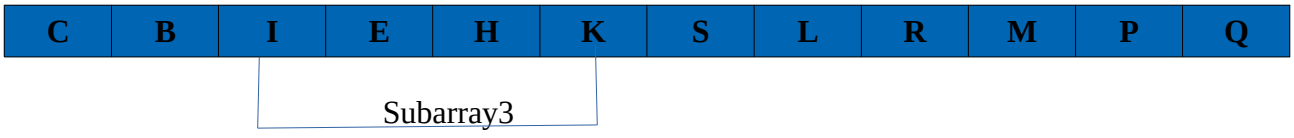
Sorting Subarray1



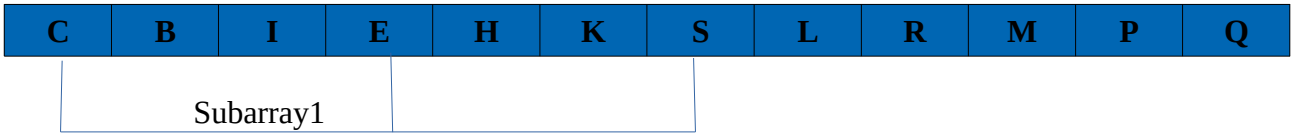
Sorting Subarray2



Sorting Subarray3

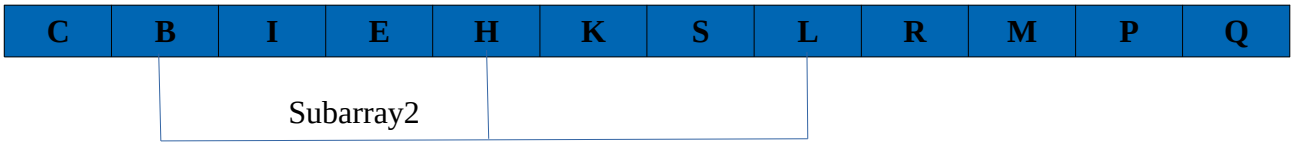


Sorting Subarray1



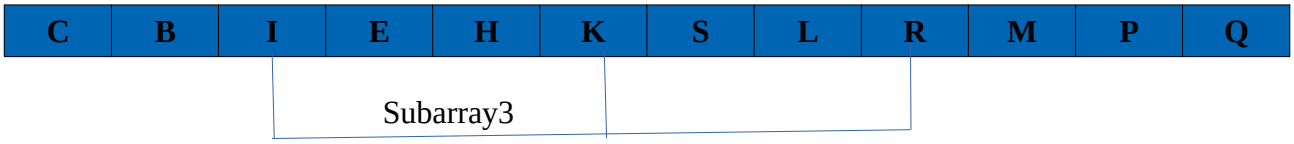
Subarray1

Sorting Subarray2



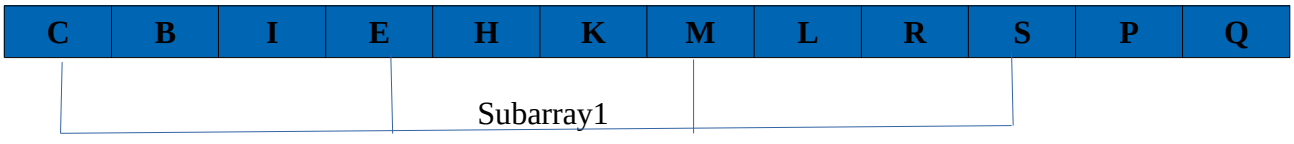
Subarray2

Sorting Subarray3



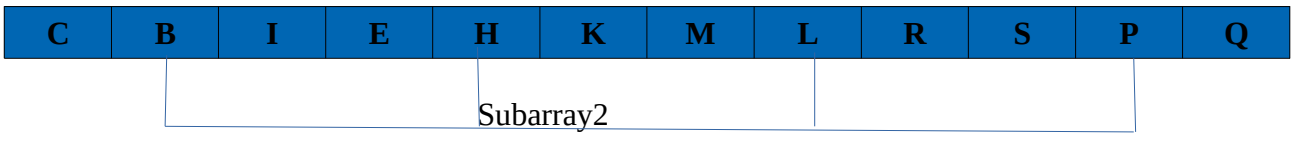
Subarray3

Sorting Subarray1



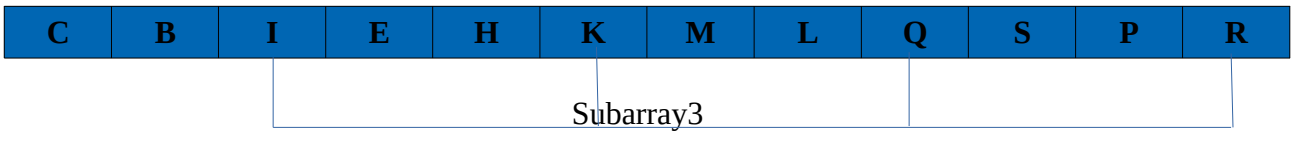
Subarray1

Sorting Subarray2



Subarray2

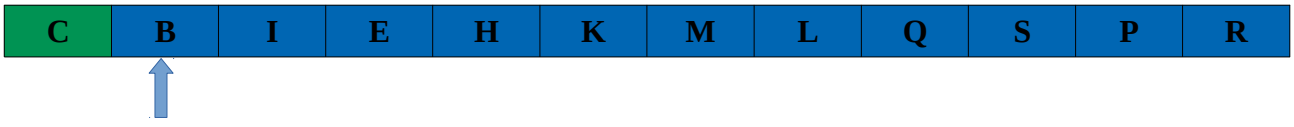
Sorting Subarray3

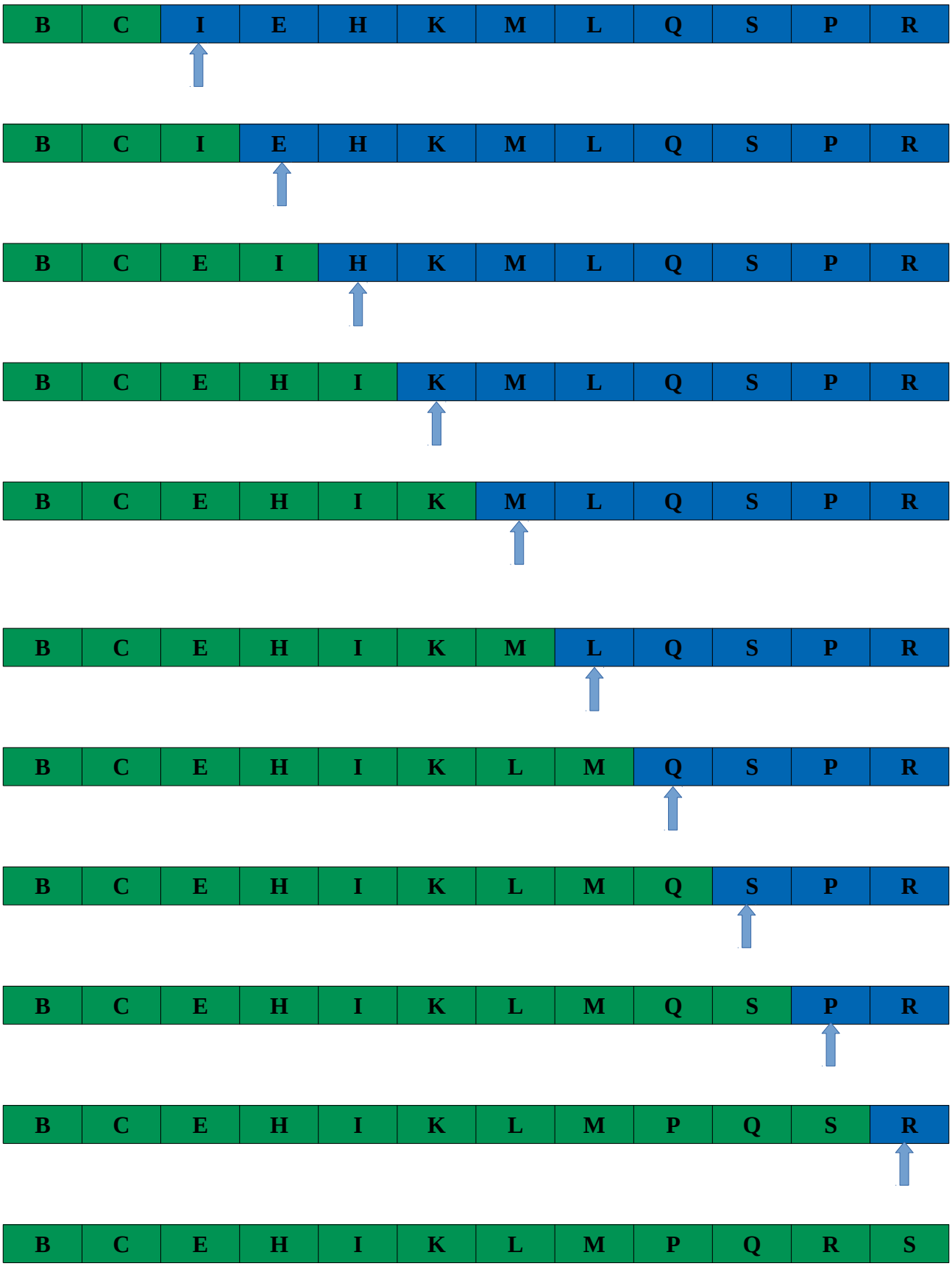


Subarray3

Sort on gap value of 1(a regular insertion sort)

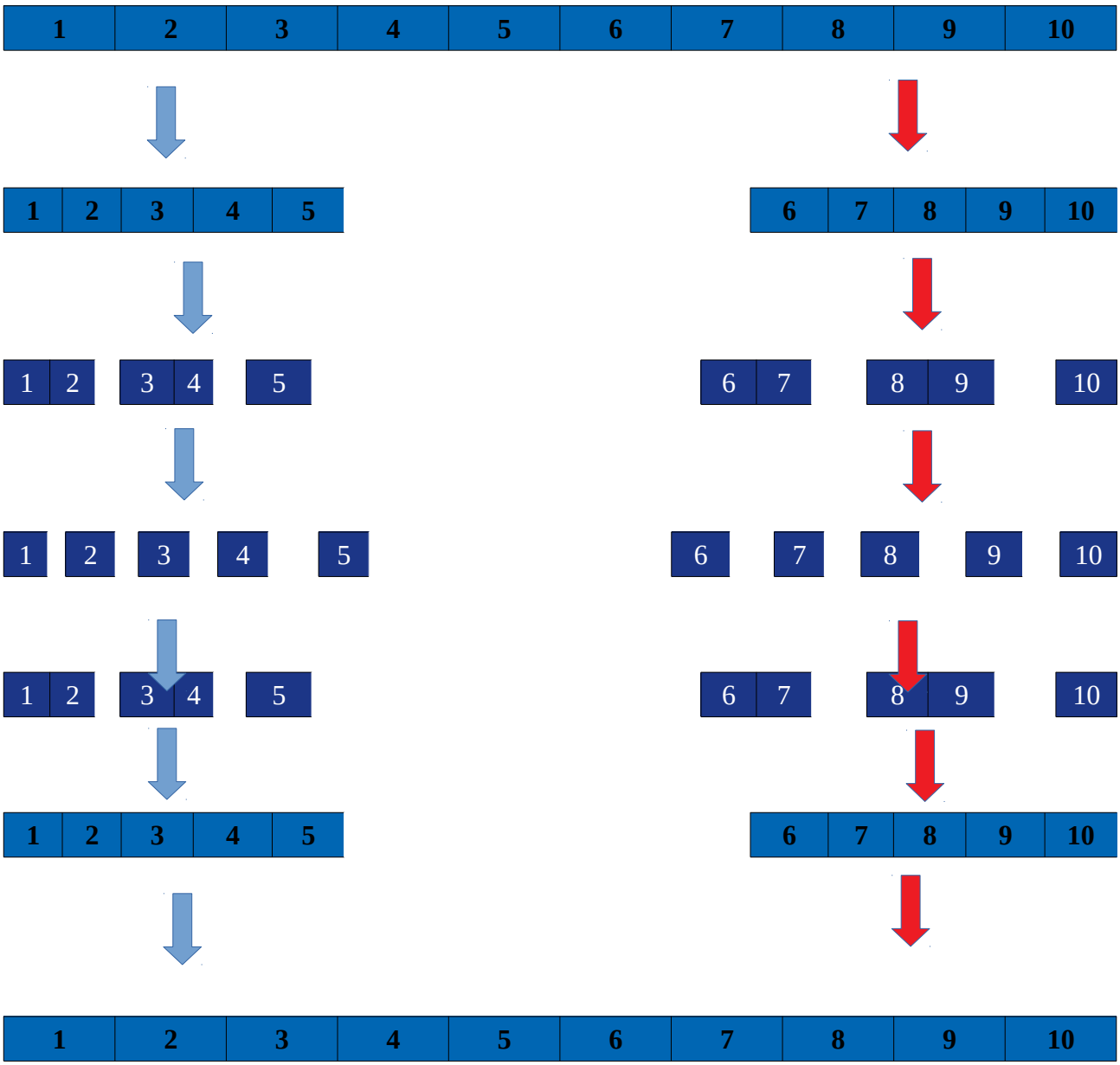
Gap value	**1**



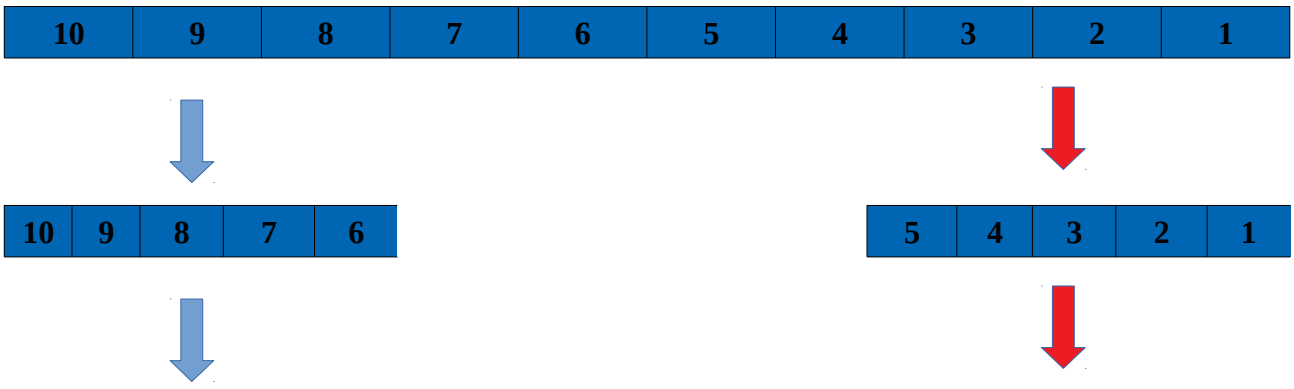


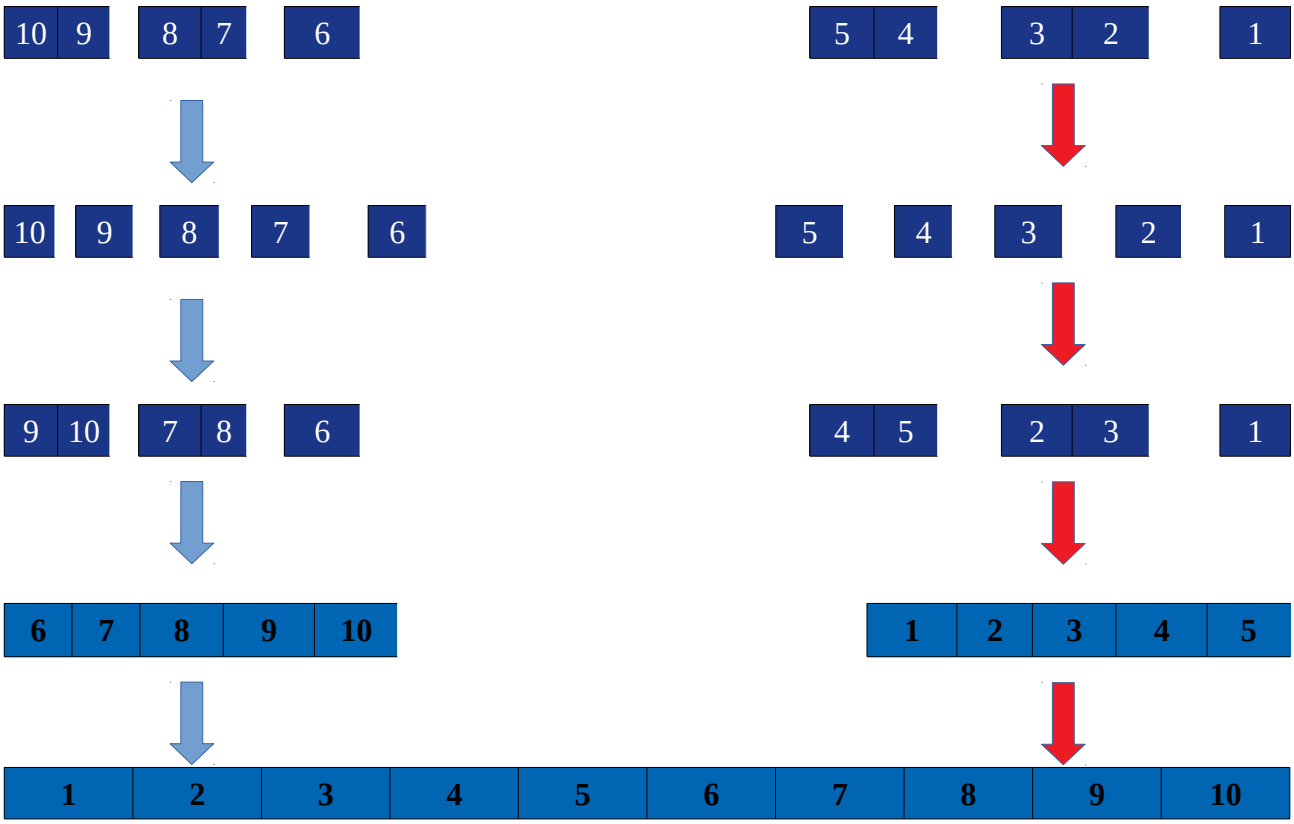
#Merge Sort#

*A={1,2,3,4,5,6,7,8,9,10}

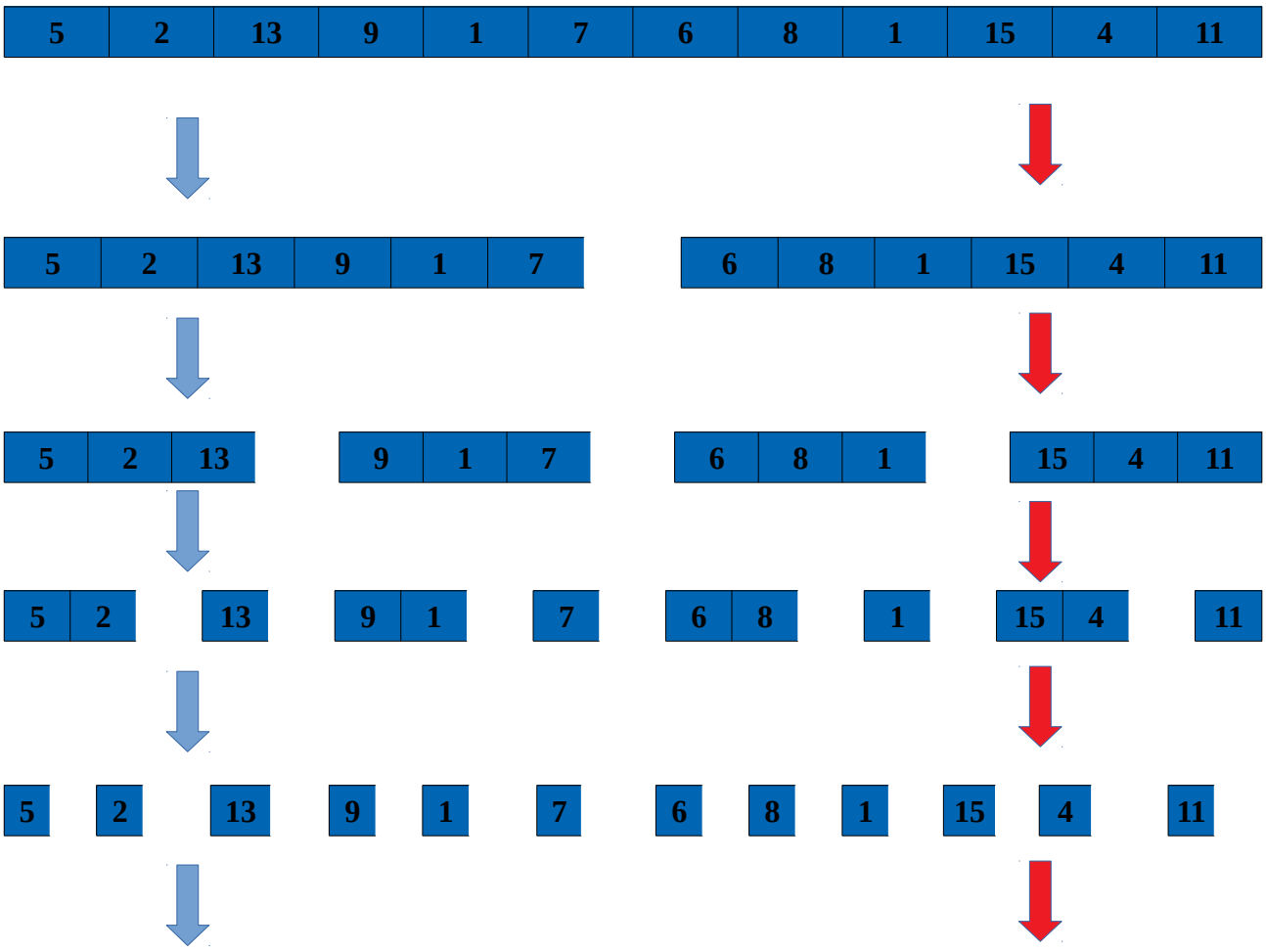


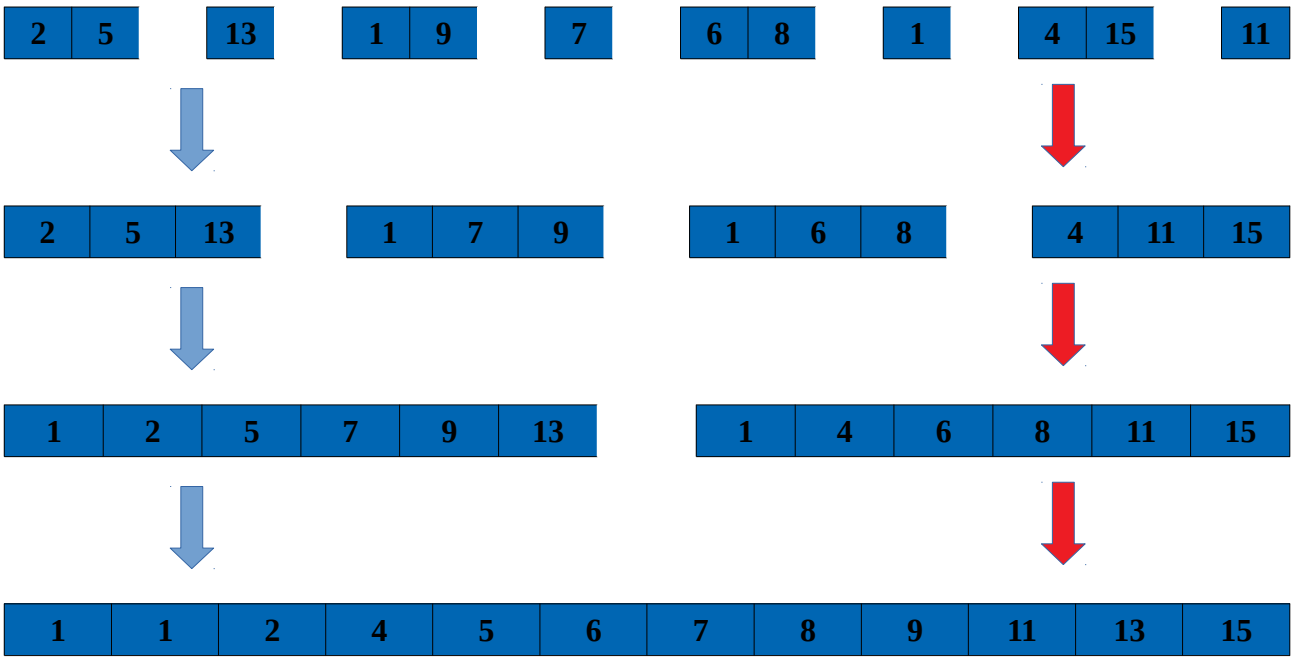
*B={10,9,8,7,6,5,4,3,2,1}



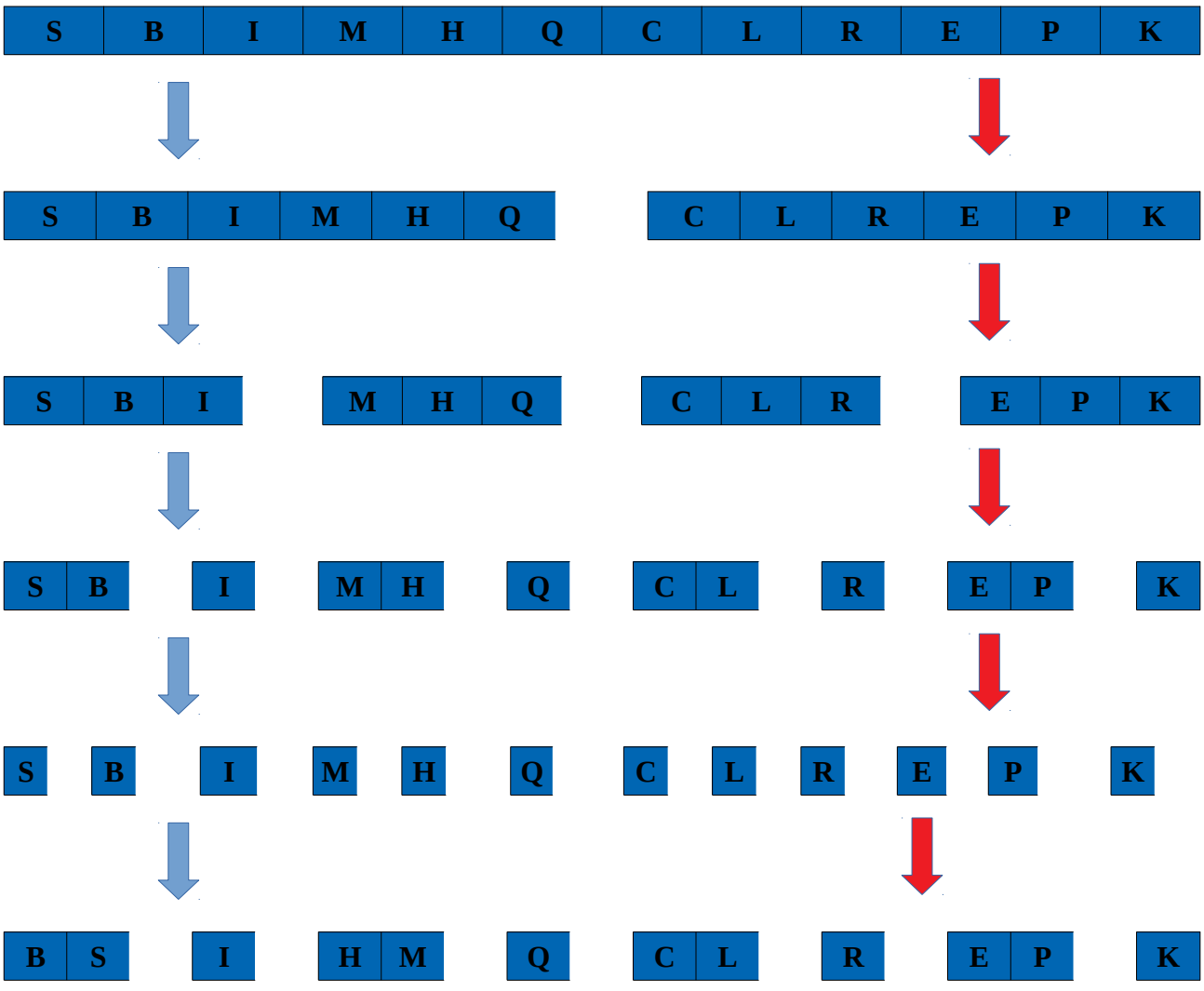


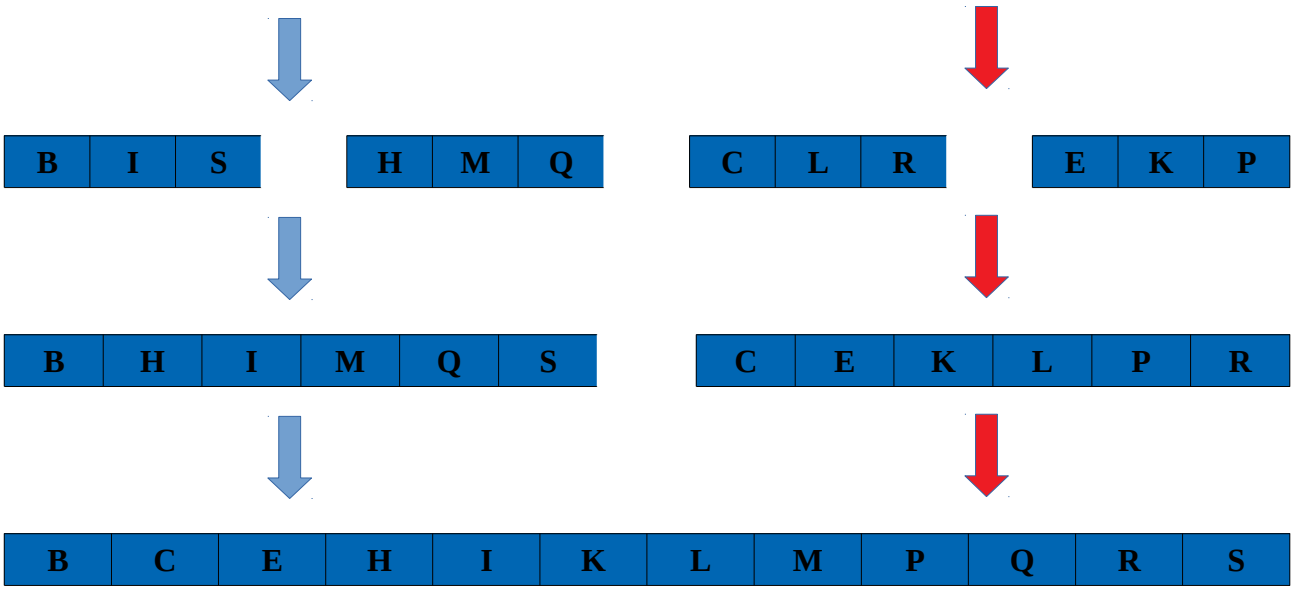
*C={5,2,13,9,1,7,6,8,1,15,4,11}





*D={‘S’,’B’,’I’,’M’,’H’,’Q’,’C’,’L’,’R’,’E’,’P’,’K’}

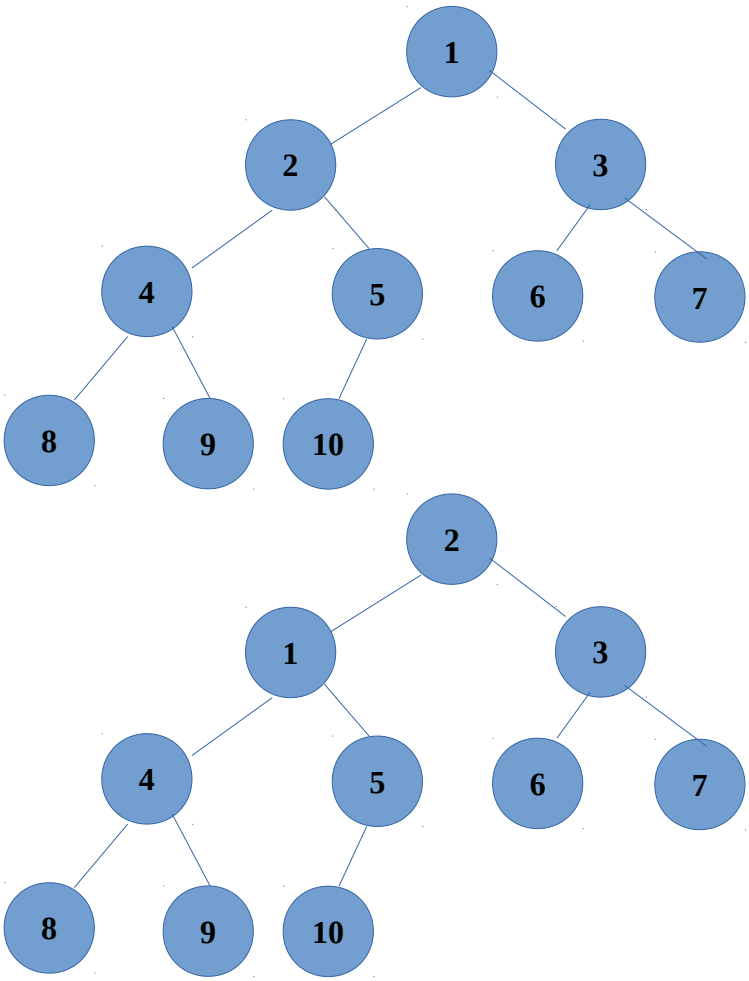


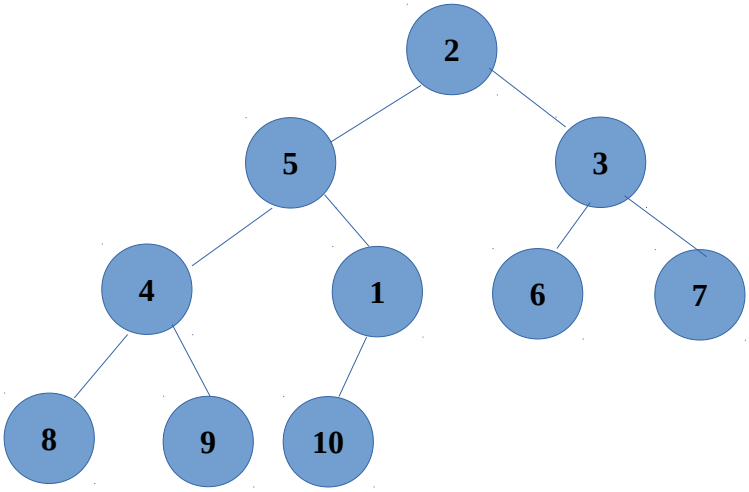


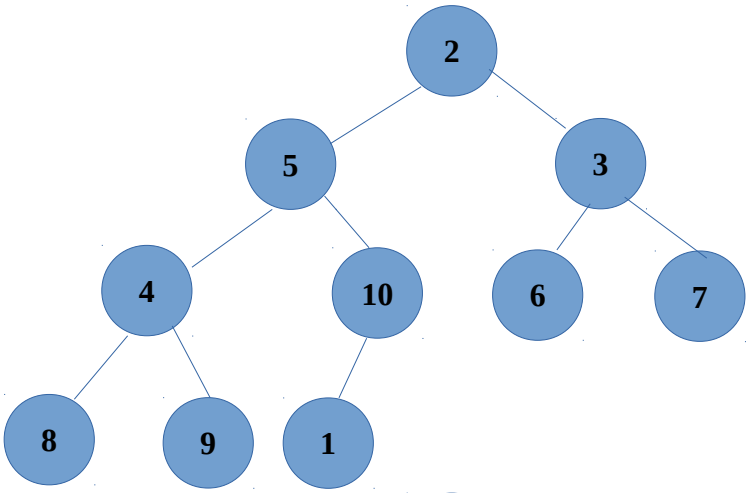
#Heap Sort#

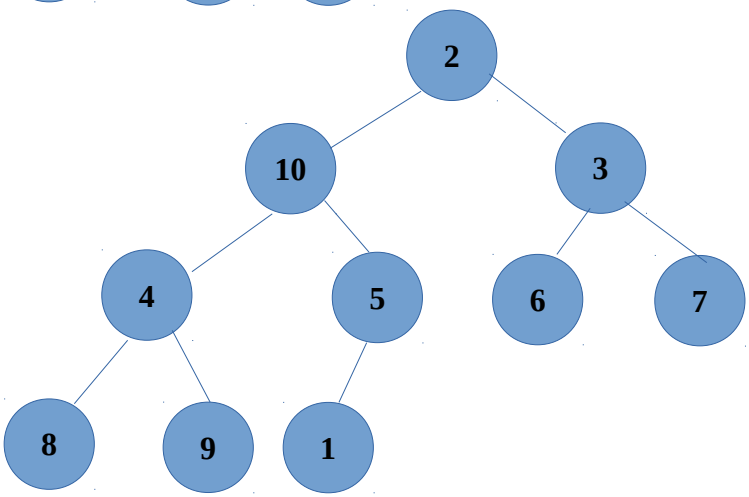
*A={1,2,3,4,5,6,7,8,9,10}

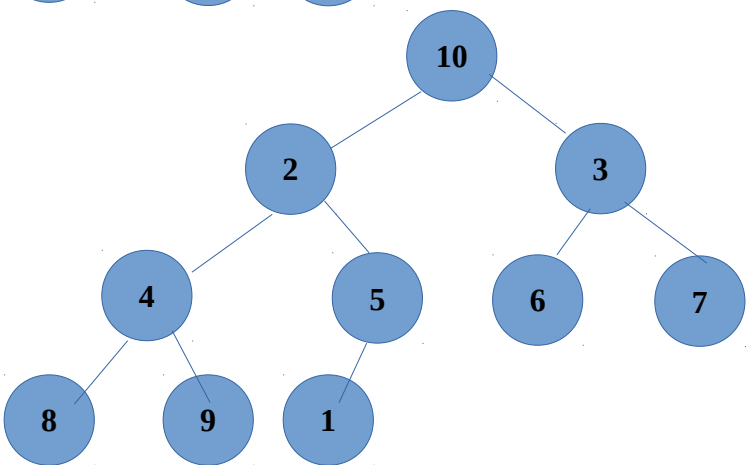
Construct heap following list of unsorted numbers

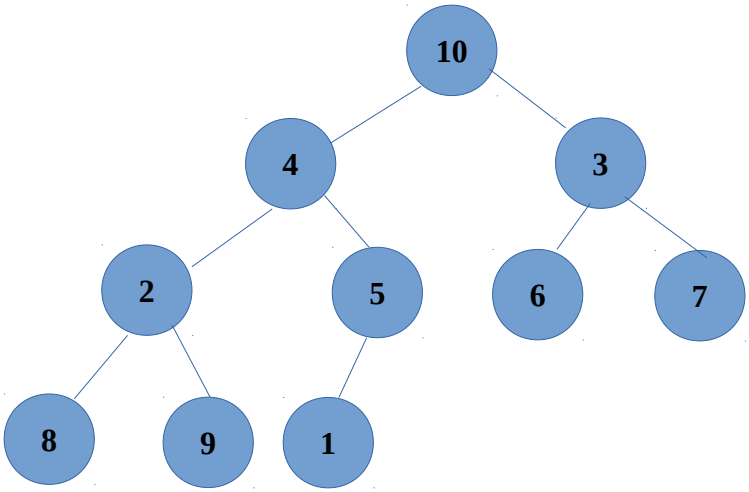


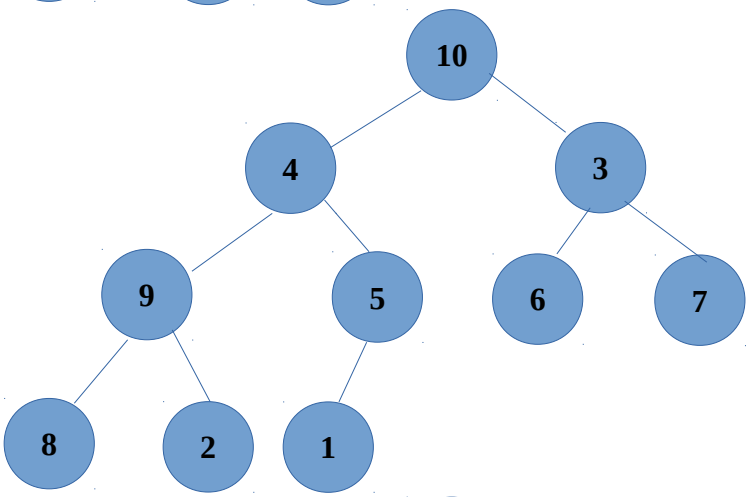

2
5
3
4
1
6
7
8
9
10

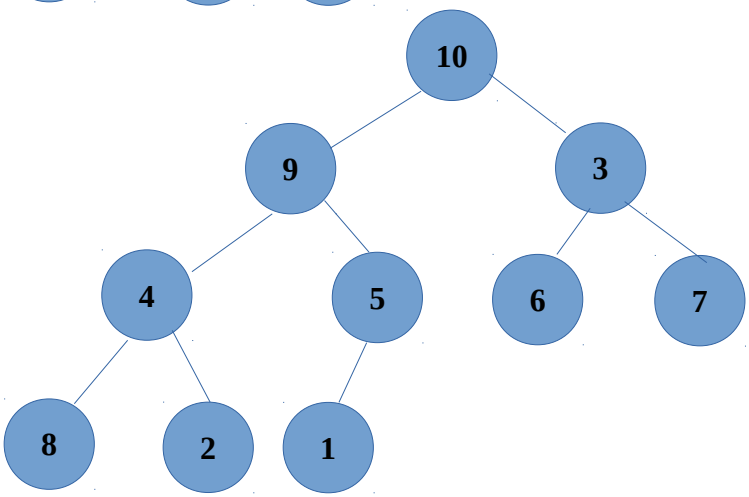

2
5
3
4
10
6
7
8
9
1

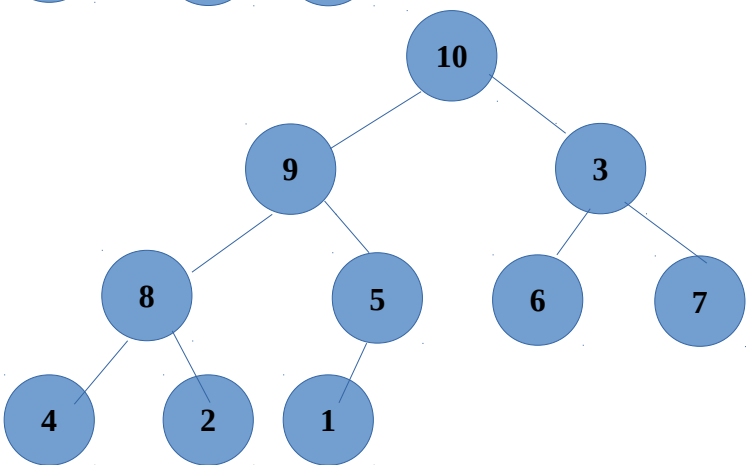

2
10
3
4
5
6
7
8
9
1

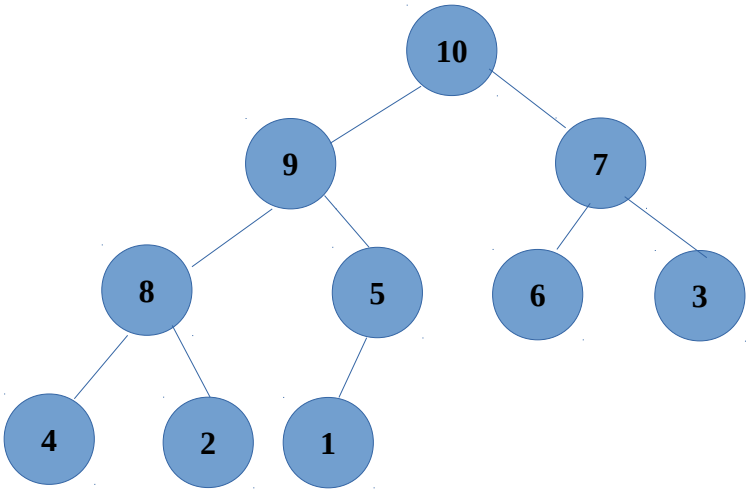

10
2
3
4
5
6
7
8
9
1

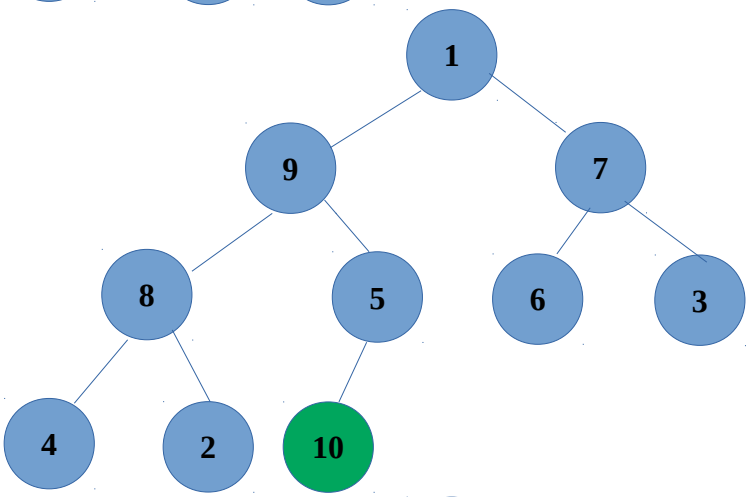

10
4
3
2
5
6
7
8
9
1

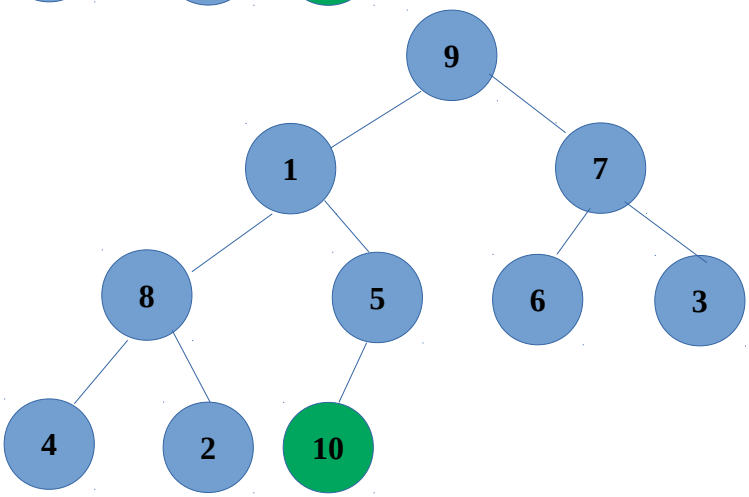

10
4
3
9
5
6
7
8
2
1

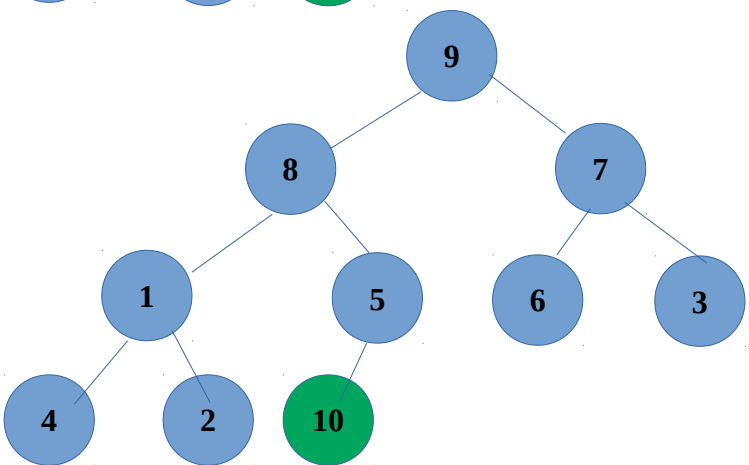

10
9
3
4
5
6
7
8
2
1

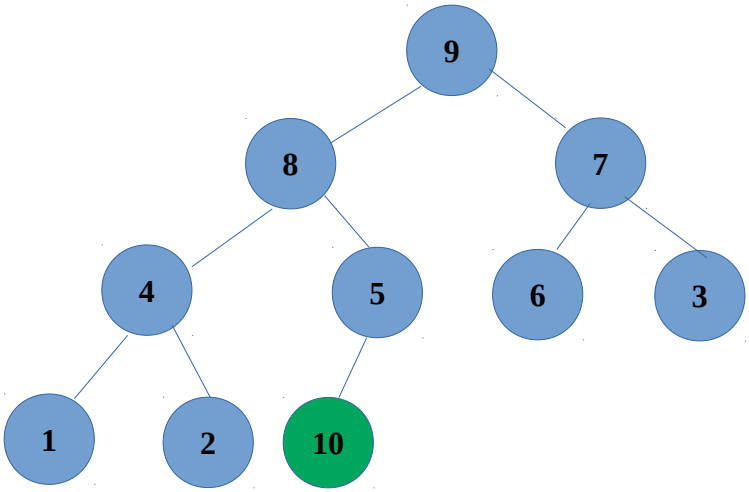

10
9
3
8
5
6
7
4
2
1

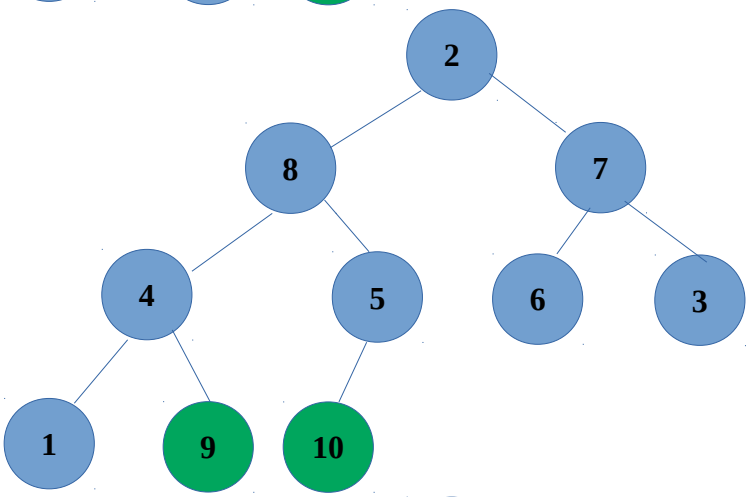

10
9
7
8
5
6
3
4
2
1

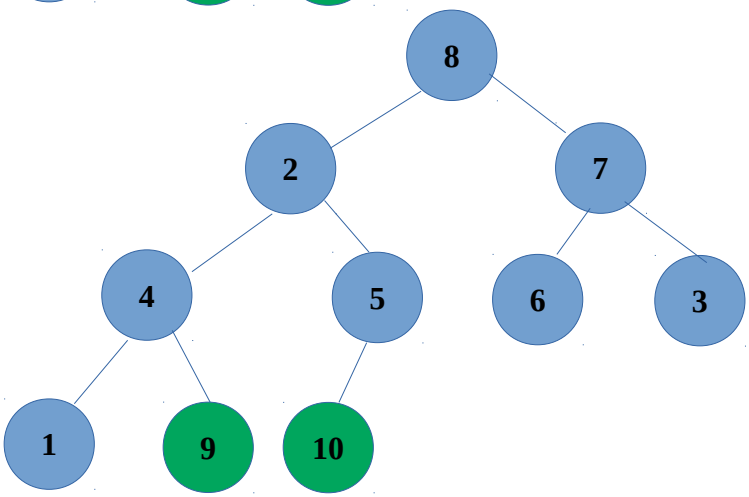

1
9
7
8
5
6
3
4
2
10

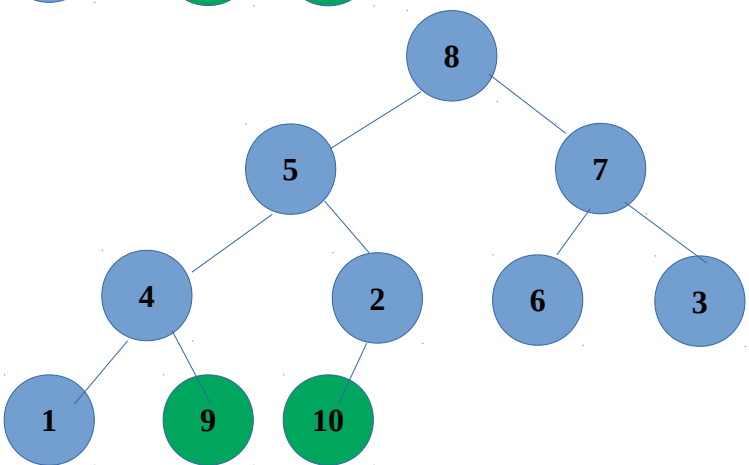

9
1
7
8
5
6
3
4
2
10

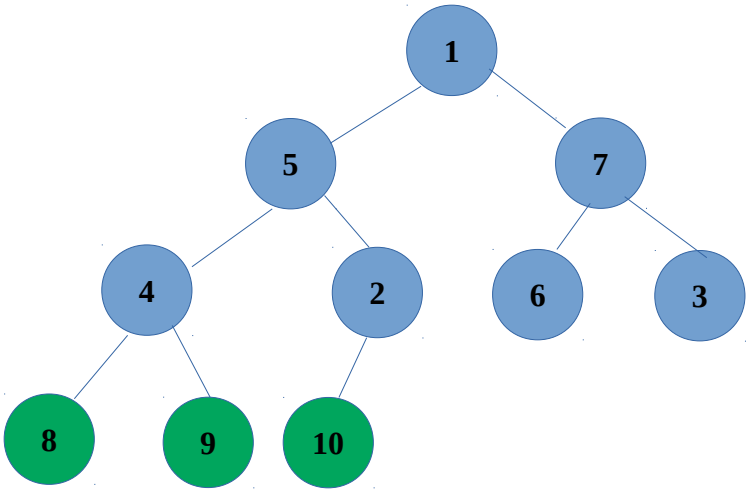

9
8
7
1
5
6
3
4
2
10

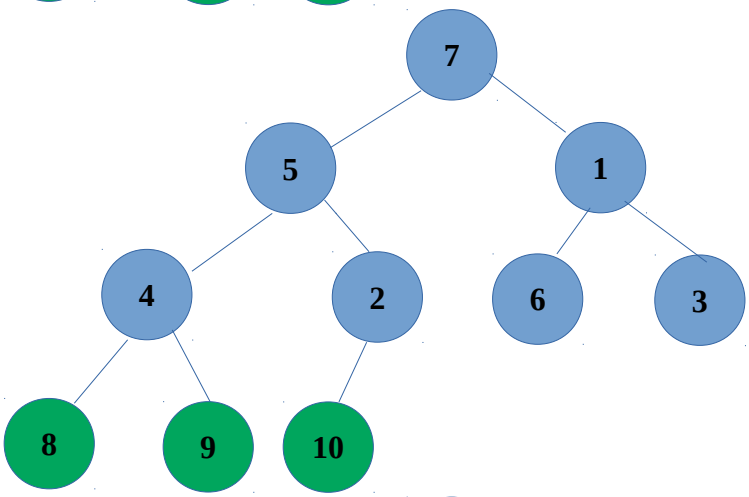

9
8
7
4
5
6
3
1
2
10

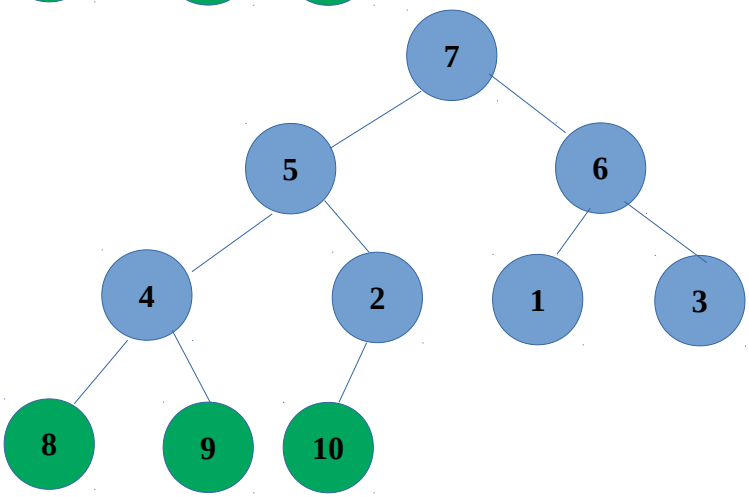

2
8
7
4
5
6
3
1
9
10

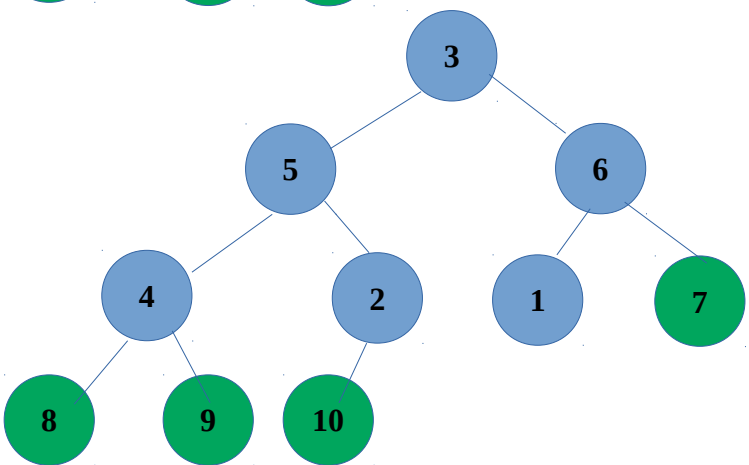

8
2
7
4
5
6
3
1
9
10

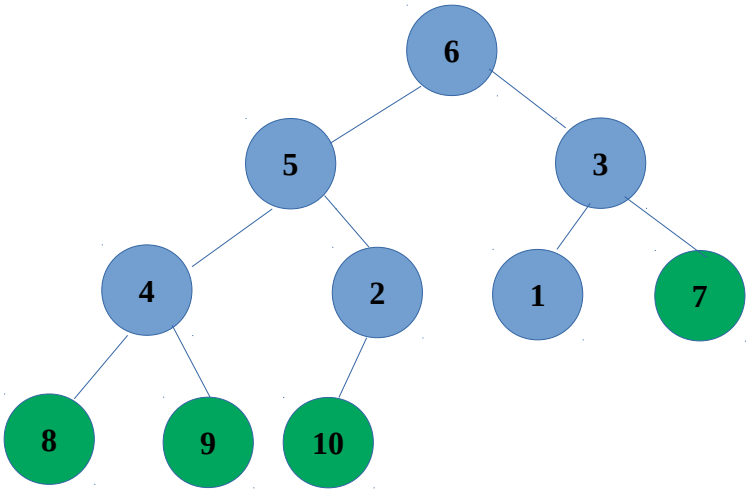

8
5
7
4
2
6
3
1
9
10

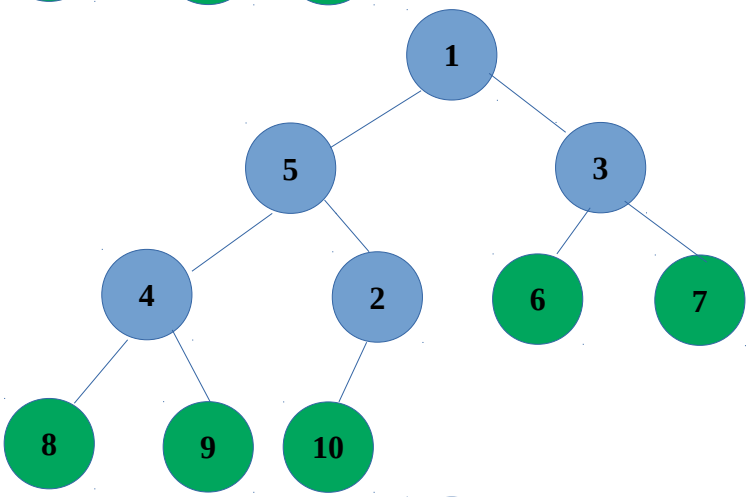

1
5
7
4
2
6
3
8
9
10

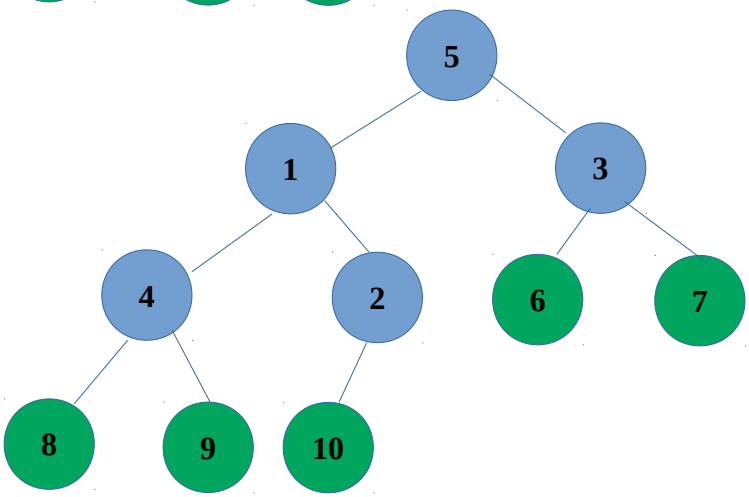

7
5
1
4
2
6
3
8
9
10

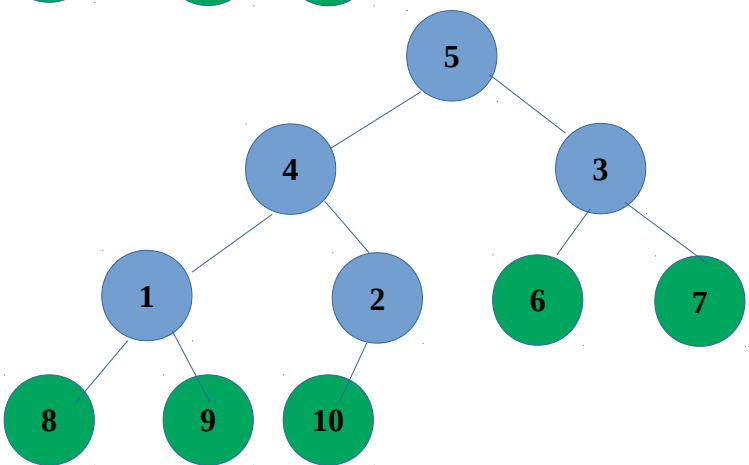

7
5
6
4
2
1
3
8
9
10

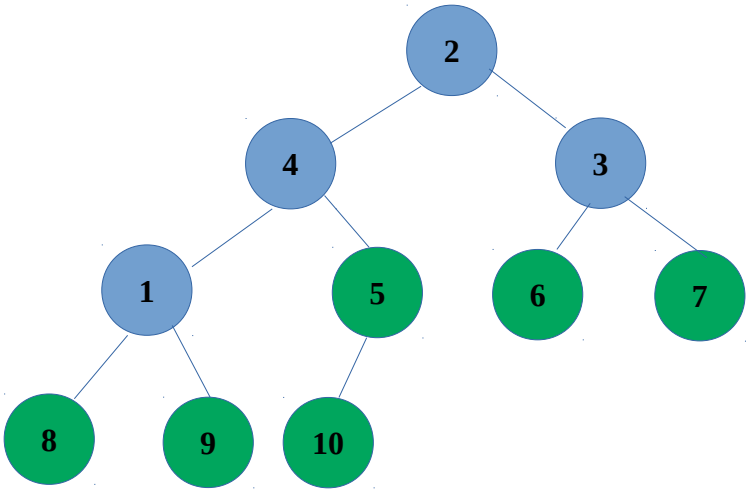

3
5
6
4
2
1
7
8
9
10

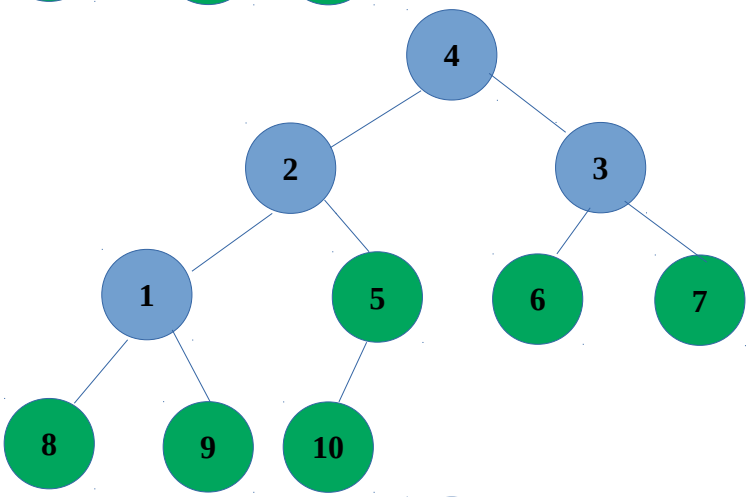

6
5
3
4
2
1
7
8
9
10

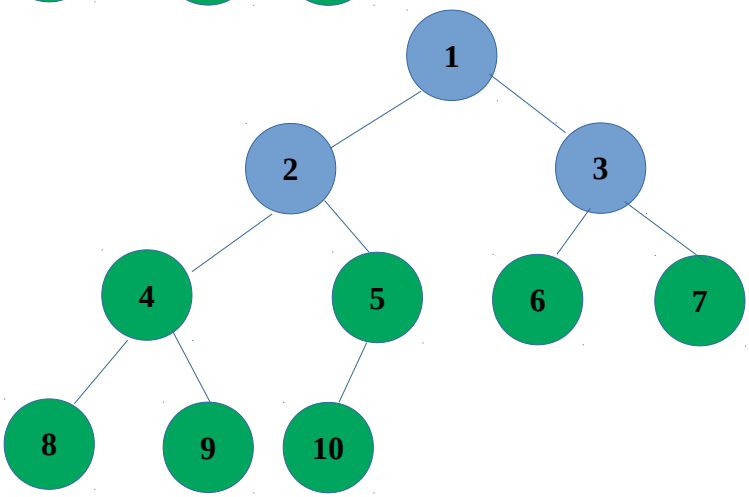

1
5
3
4
2
6
7
8
9
10

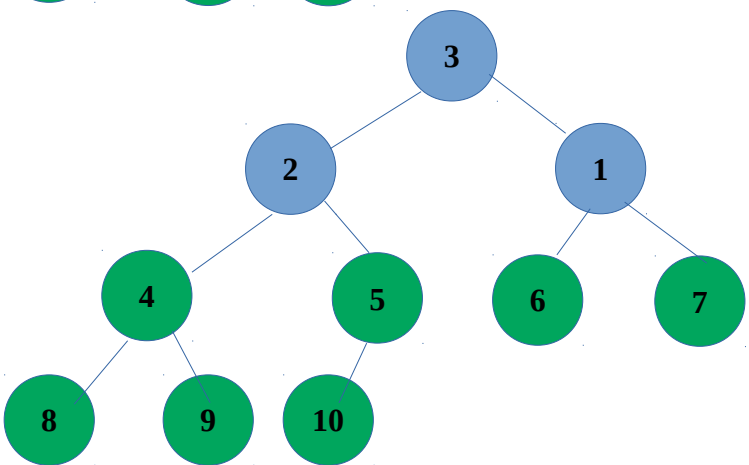

5
1
3
4
2
6
7
8
9
10

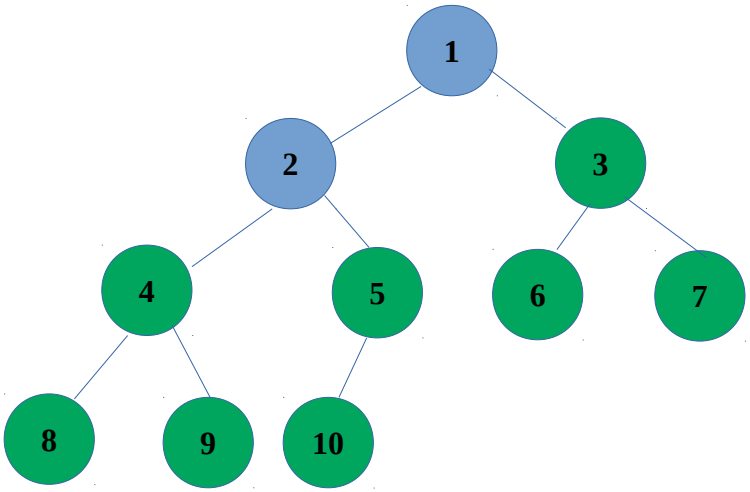

5
4
3
1
2
6
7
8
9
10

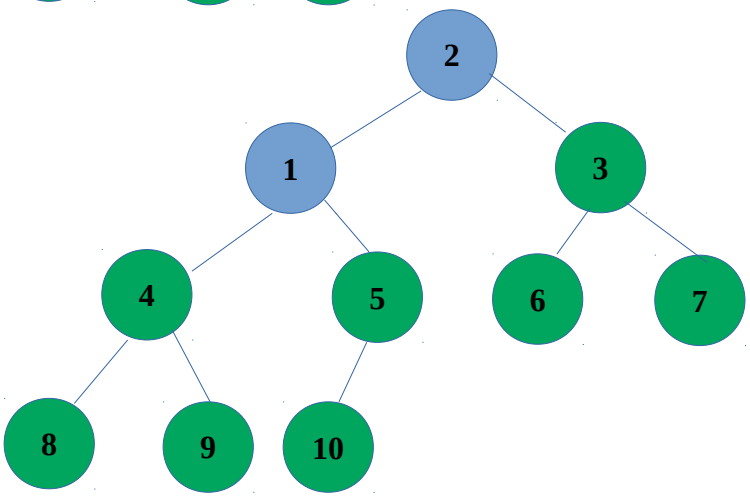

2
4
3
1
5
6
7
8
9
10

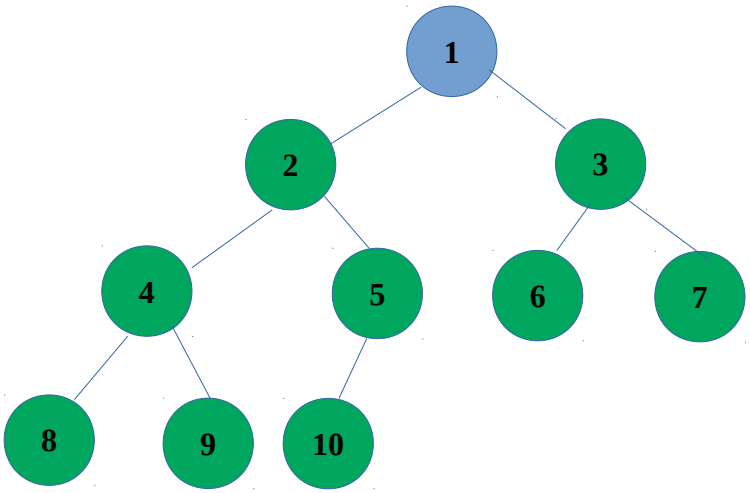

4
2
3
1
5
6
7
8
9
10

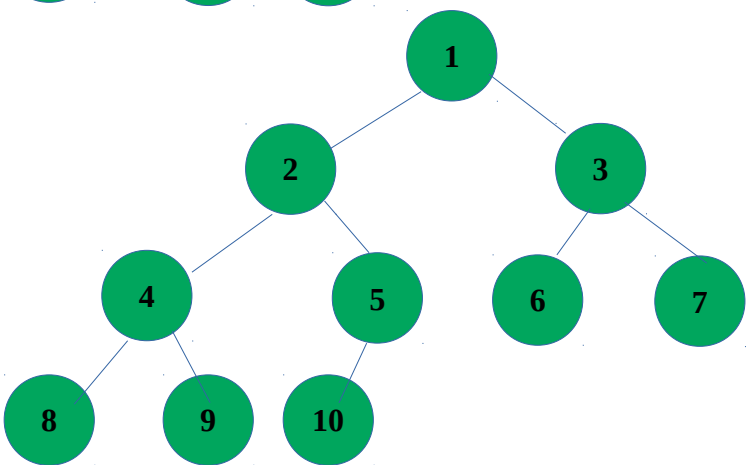

1
2
3
4
5
6
7
8
9
10


3
2
1
4
5
6
7
8
9
10


1
2
3
4
5
6
7
8
9
10

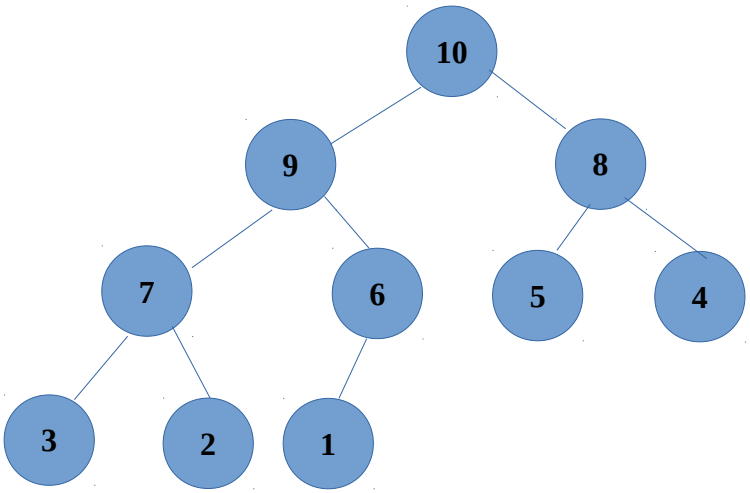

2
1
3
4
5
6
7
8
9
10

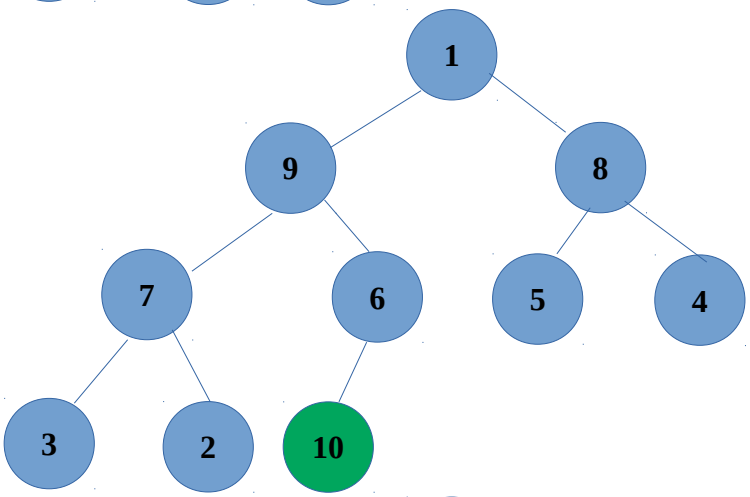

1
2
3
4
5
6
7
8
9
10

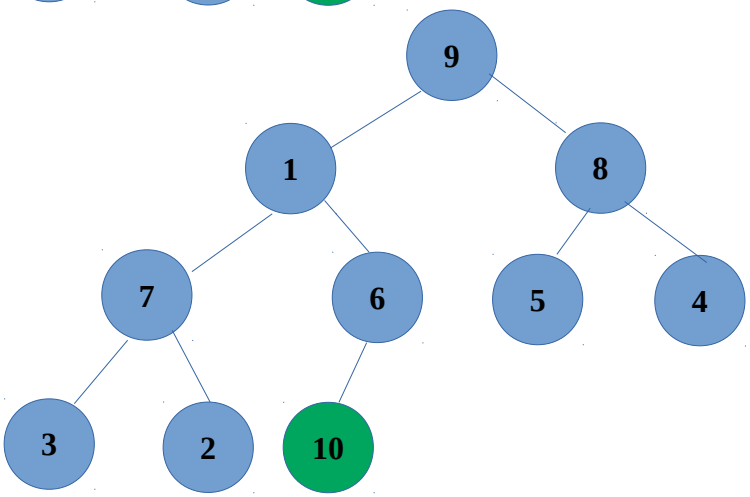

1
2
3
4
5
6
7
8
9
10

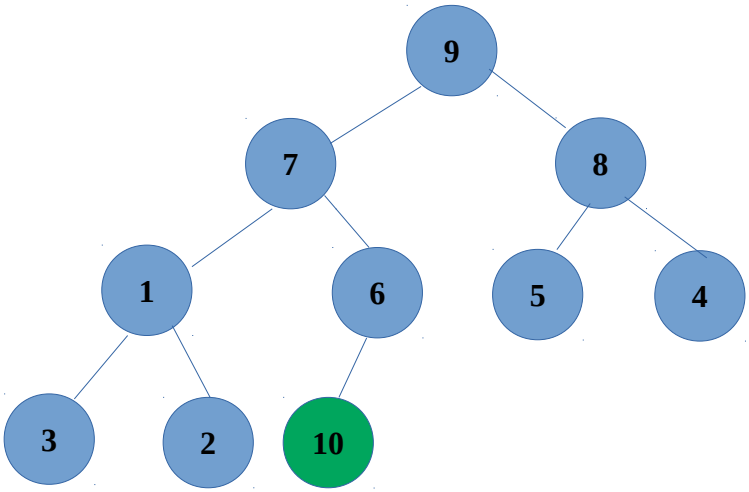
*B={10,9,8,7,6,5,4,3,2,1}

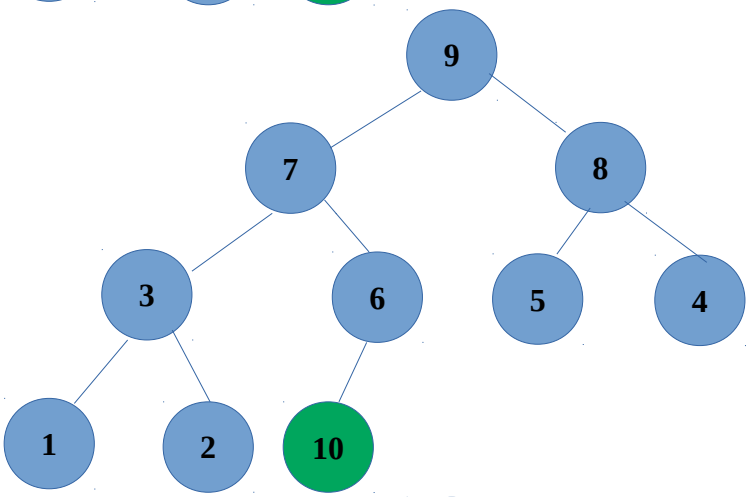
Construct heap following list of unsorted numbers

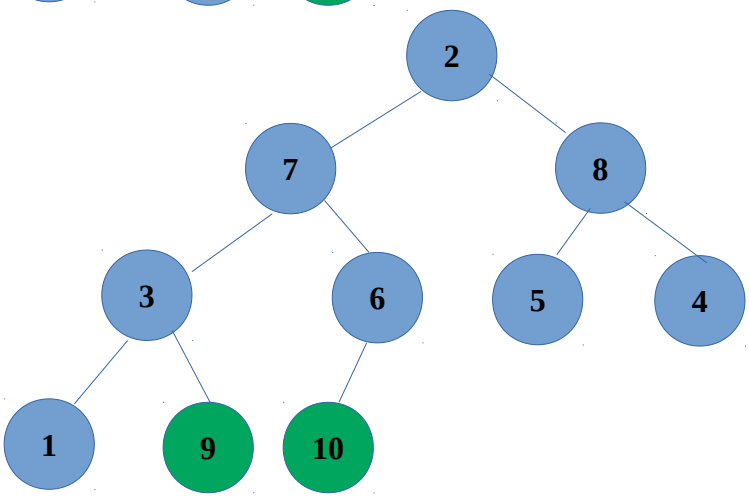


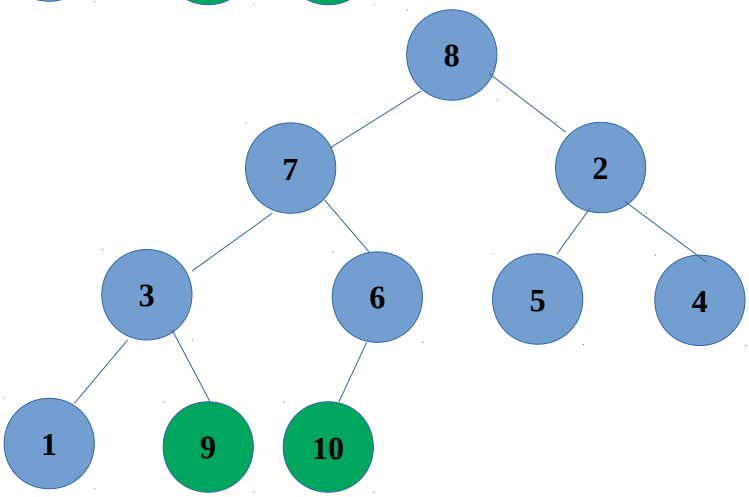


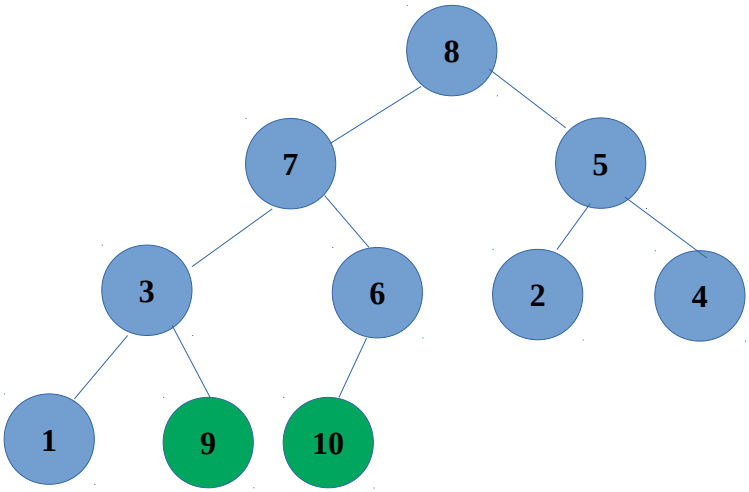


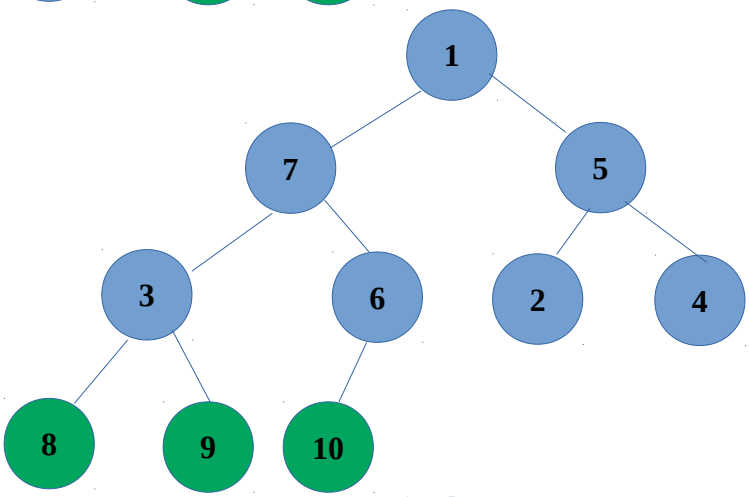

9
7
8
1
6
5
4
3
2
10

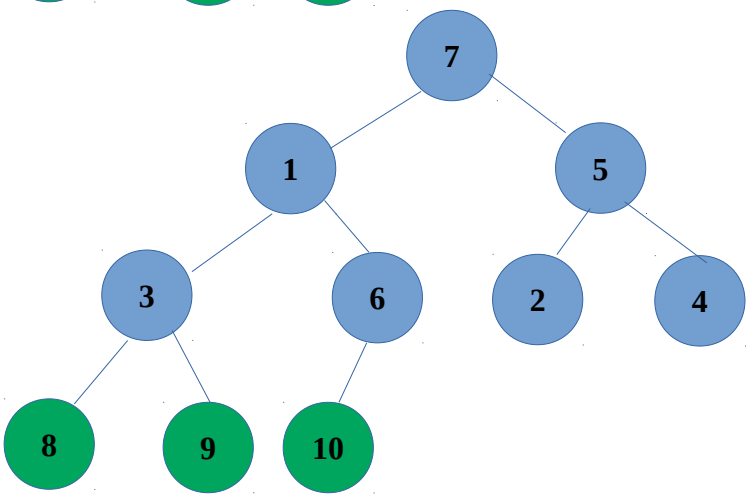

9
7
8
3
6
5
4
1
2
10

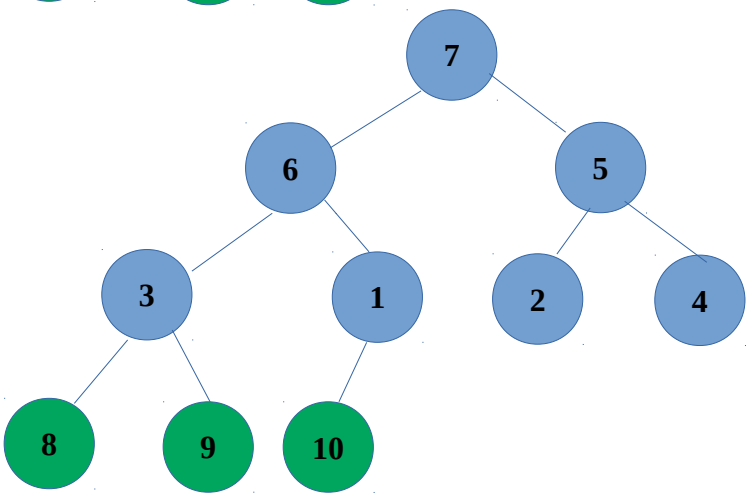

2
7
8
3
6
5
4
1
9
10

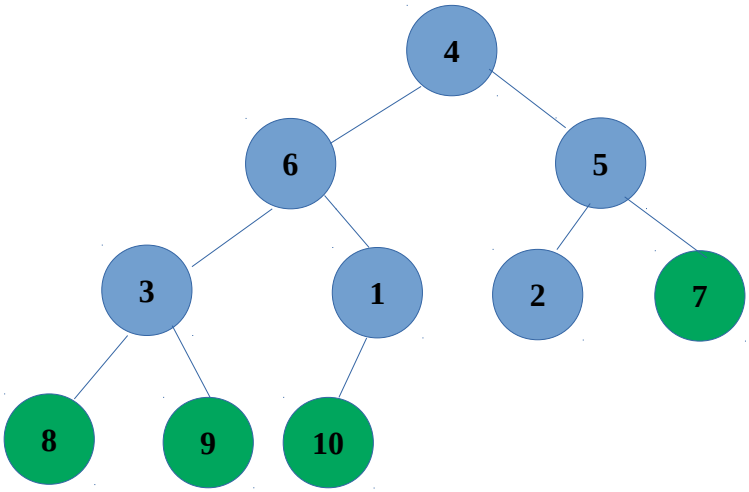

8
7
2
3
6
5
4
1
9
10

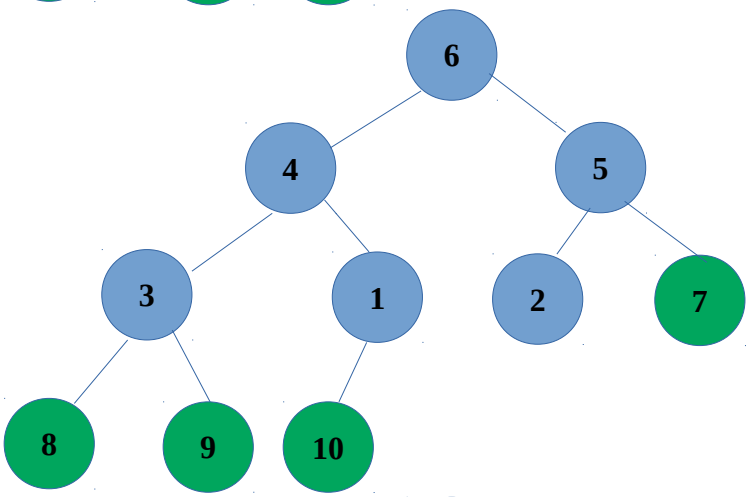

8
7
5
3
6
2
4
1
9
10

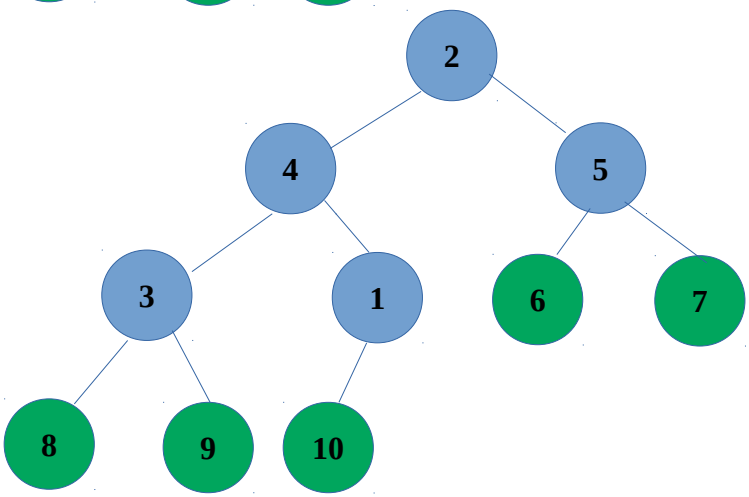

1
7
5
3
6
2
4
8
9
10

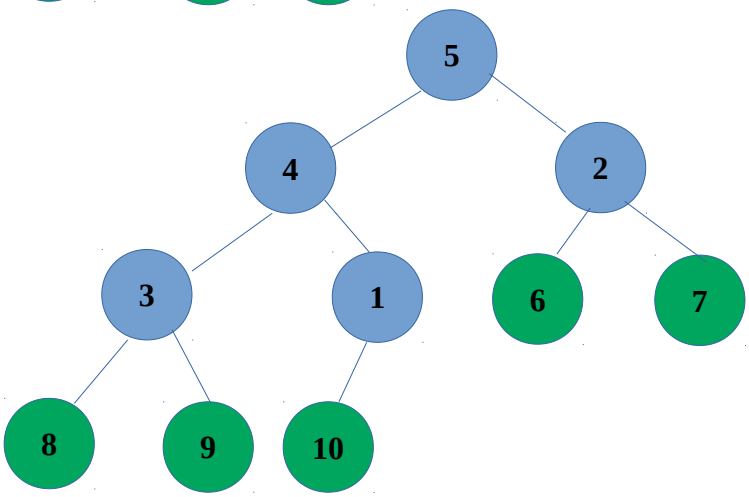

7
1
5
3
6
2
4
8
9
10

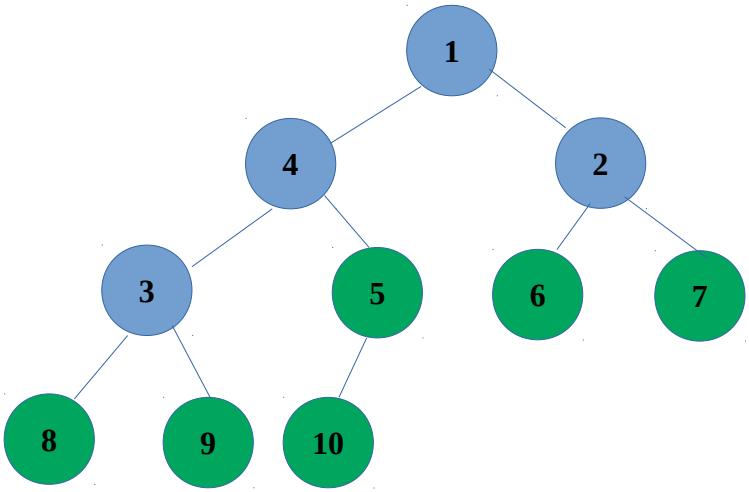

7
6
5
3
1
2
4
8
9
10

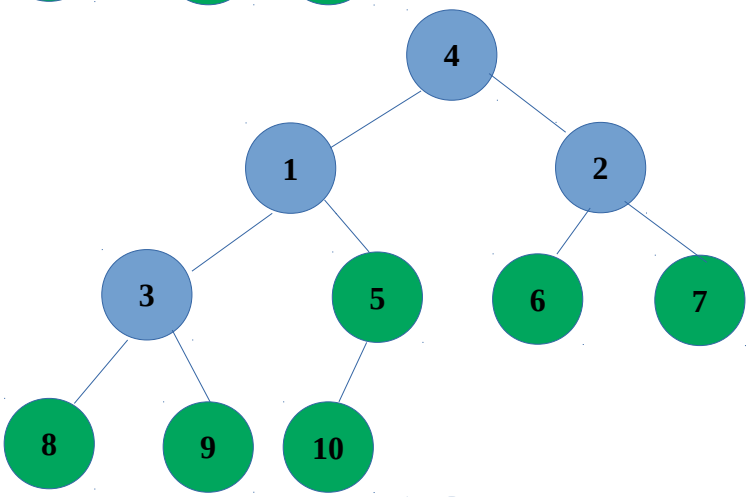

4
6
5
3
1
2
7
8
9
10

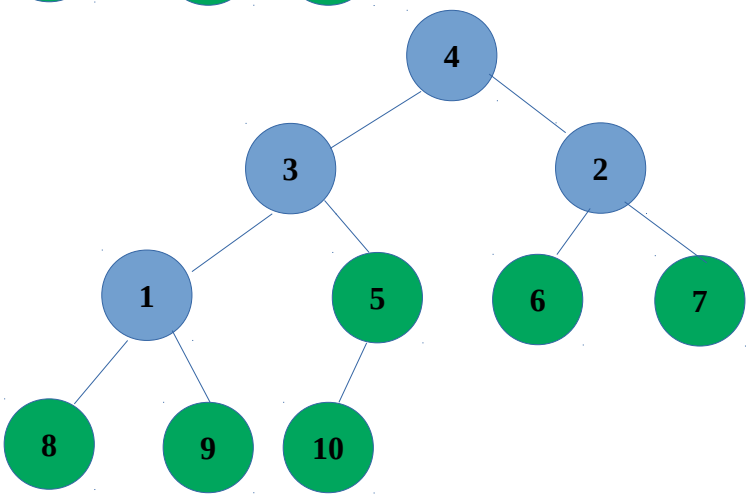

6
4
5
3
1
2
7
8
9
10

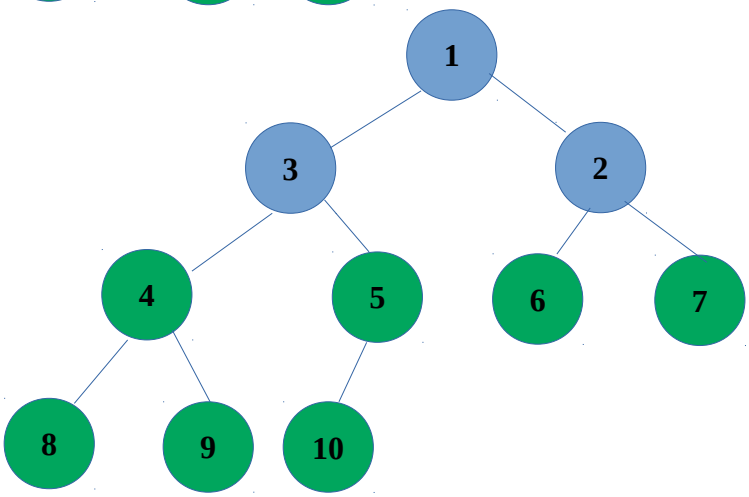

2
4
5
3
1
6
7
8
9
10

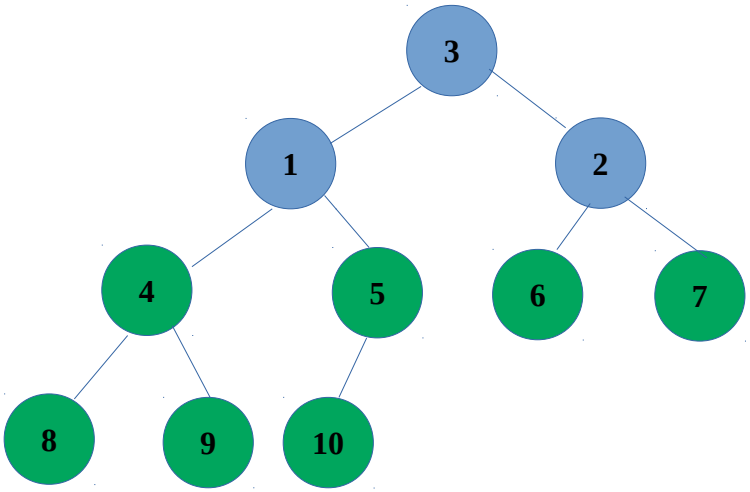

5
4
2
3
1
6
7
8
9
10

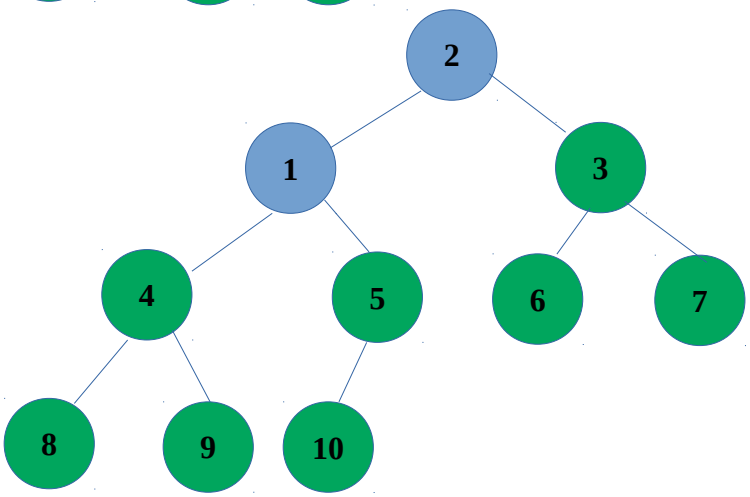

1
4
2
3
5
6
7
8
9
10

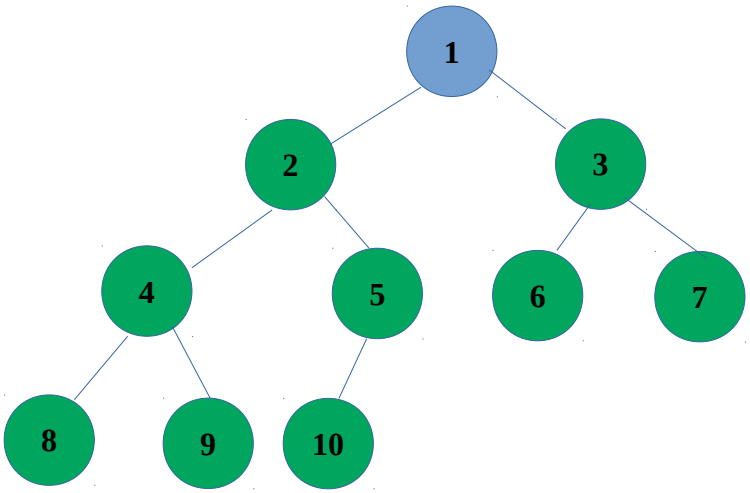

4
1
2
3
5
6
7
8
9
10

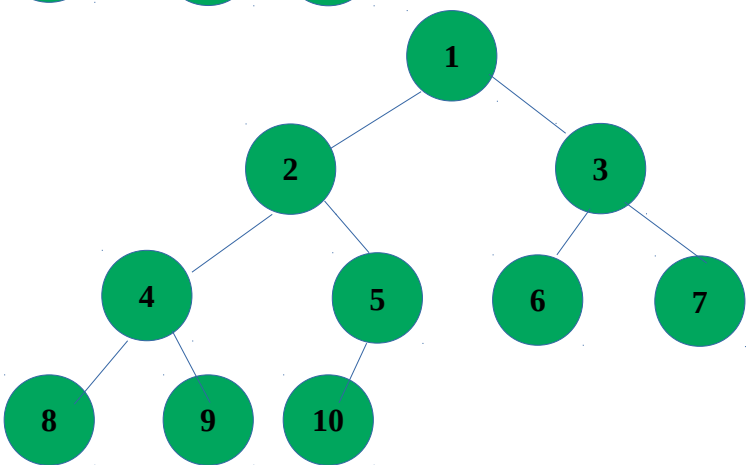

4
3
2
1
5
6
7
8
9
10


1
3
2
4
5
6
7
8
9
10


3
1
2
4
5
6
7
8
9
10

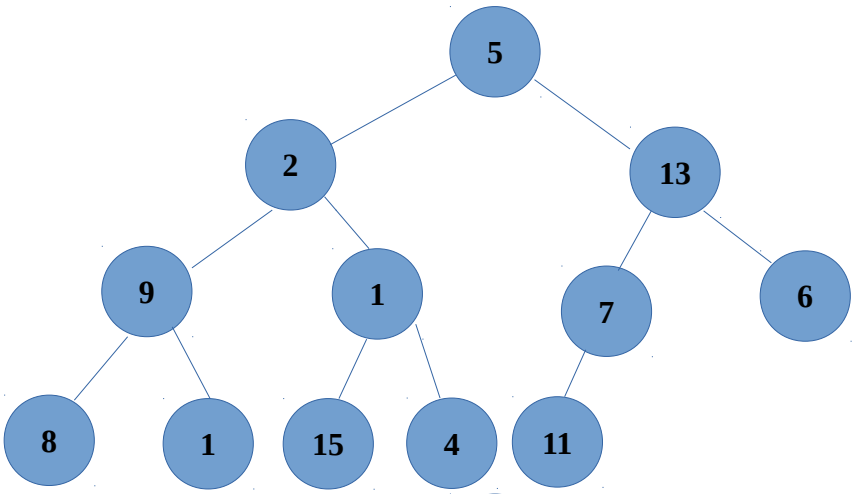

2
1
3
4
5
6
7
8
9
10

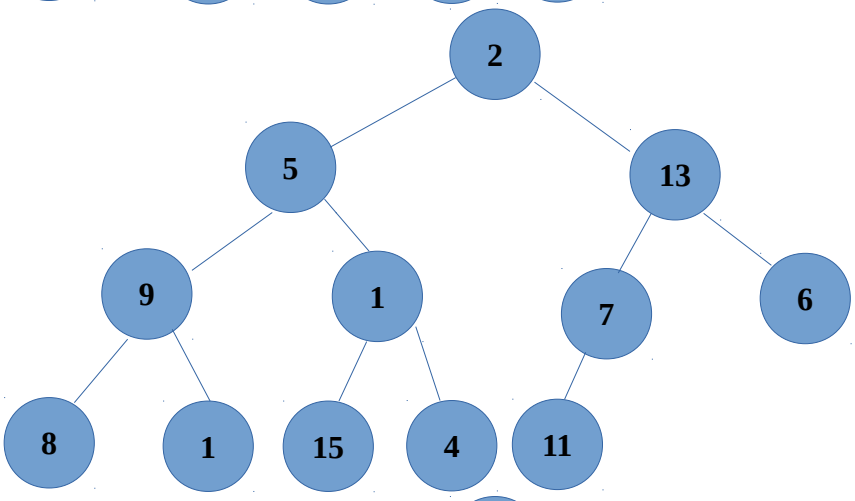

1
2
3
4
5
6
7
8
9
10

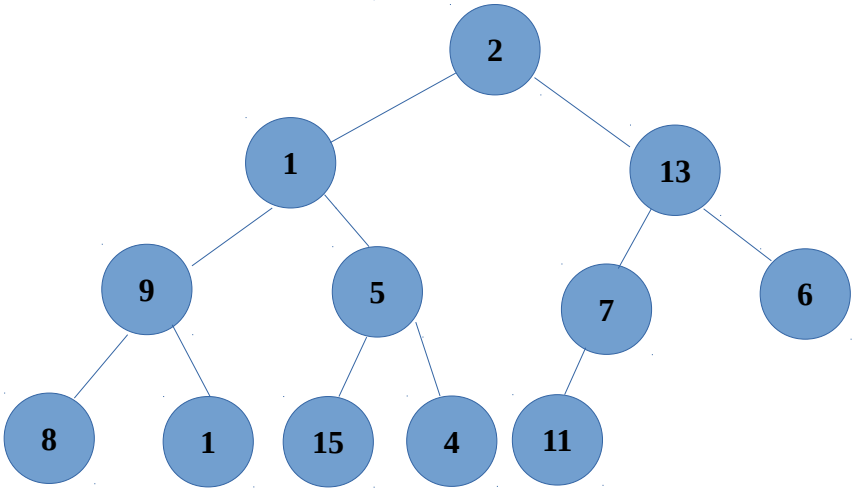

1
2
3
4
5
6
7
8
9
10

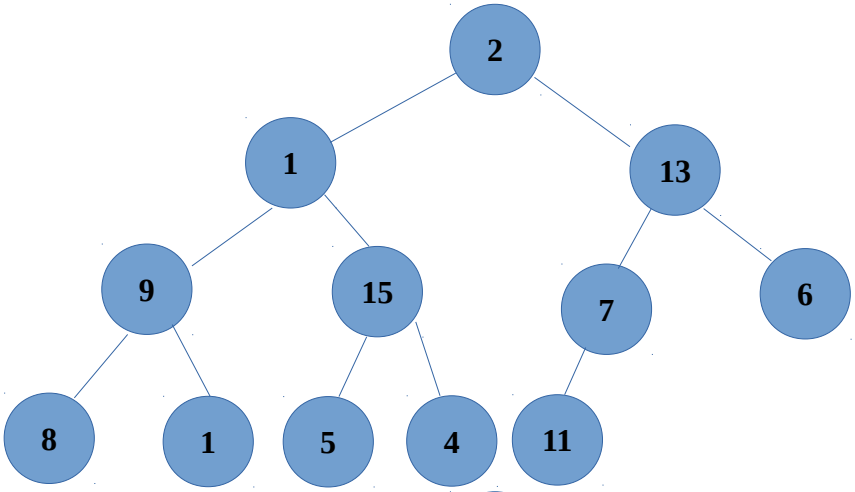
*C={5,2,13,9,1,7,6,8,1,15,4,11}

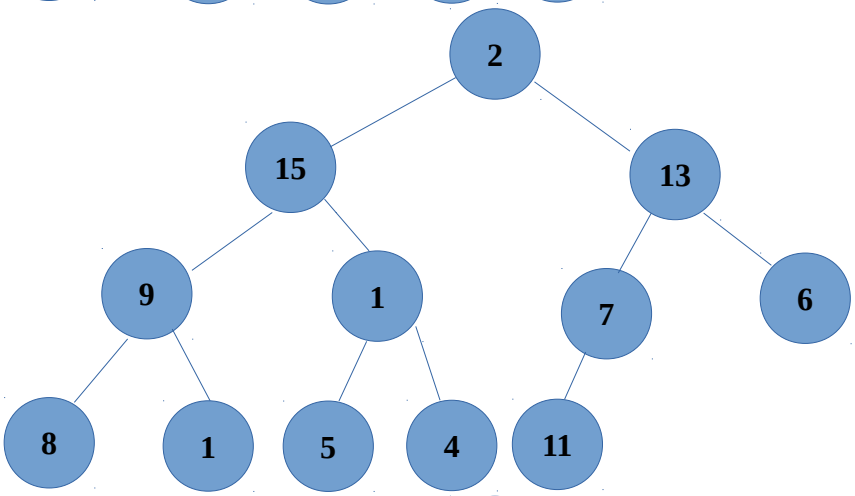
Construct heap following list of unsorted numbers

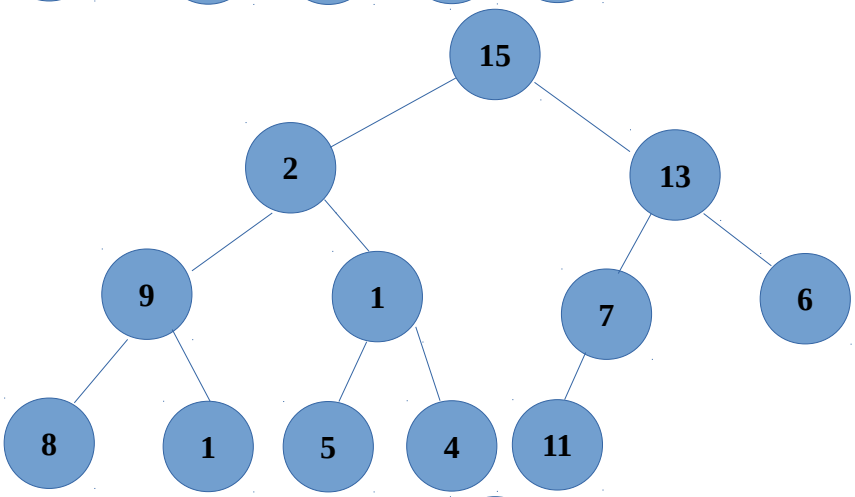


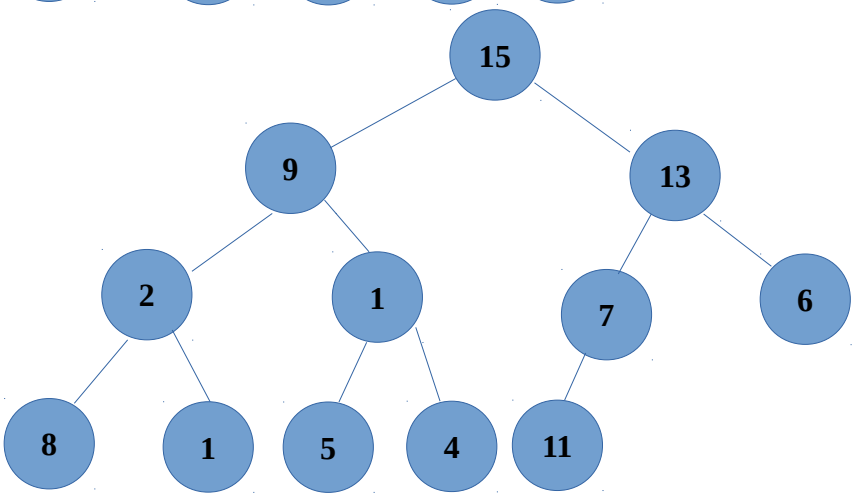


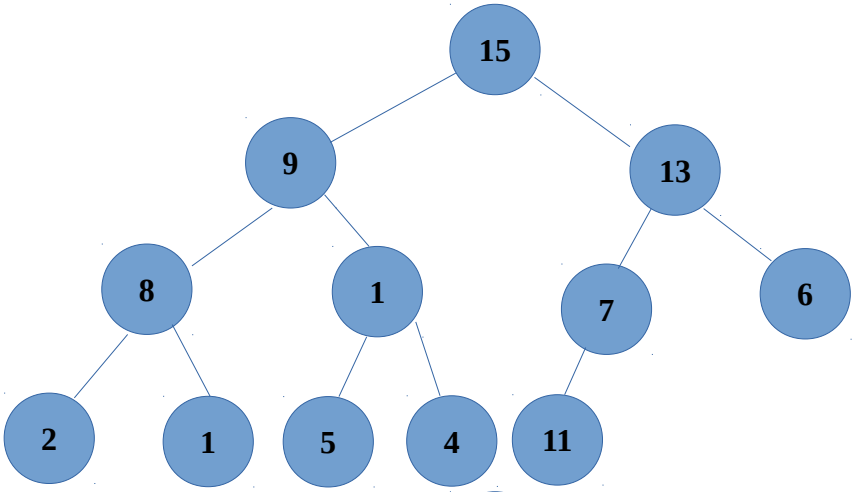


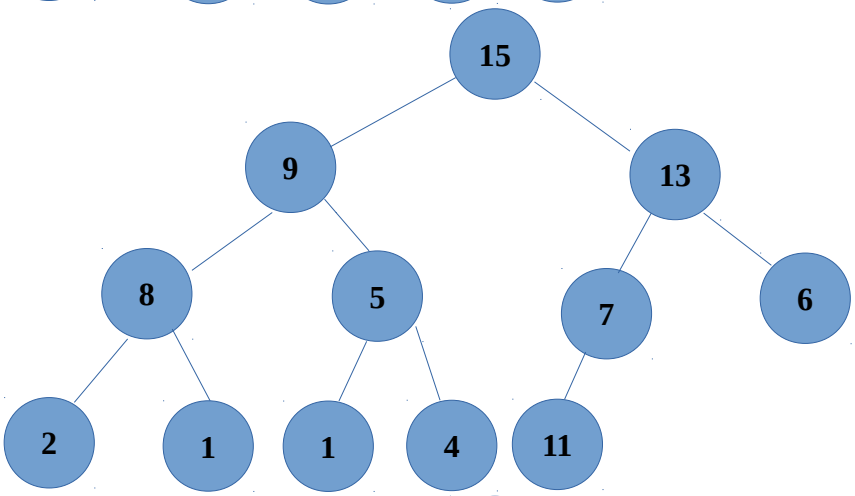

2
1
13
9
15
7
6
8
1
5
4
11

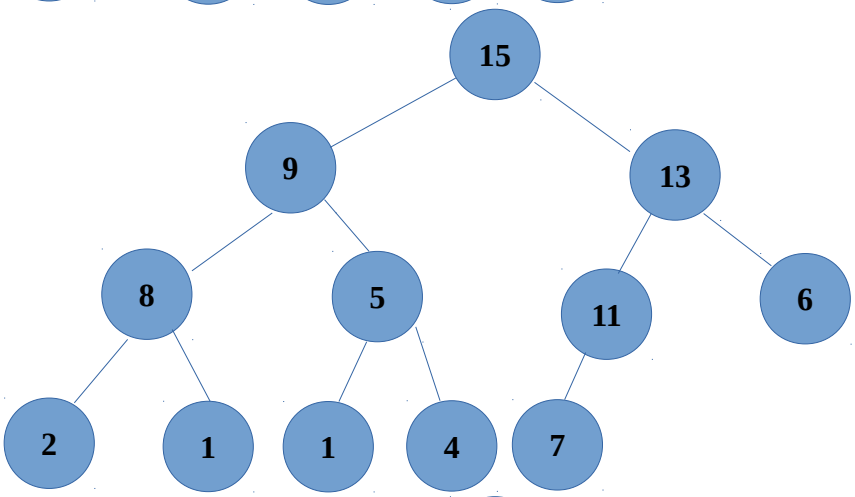

2
15
13
9
1
7
6
8
1
5
4
11

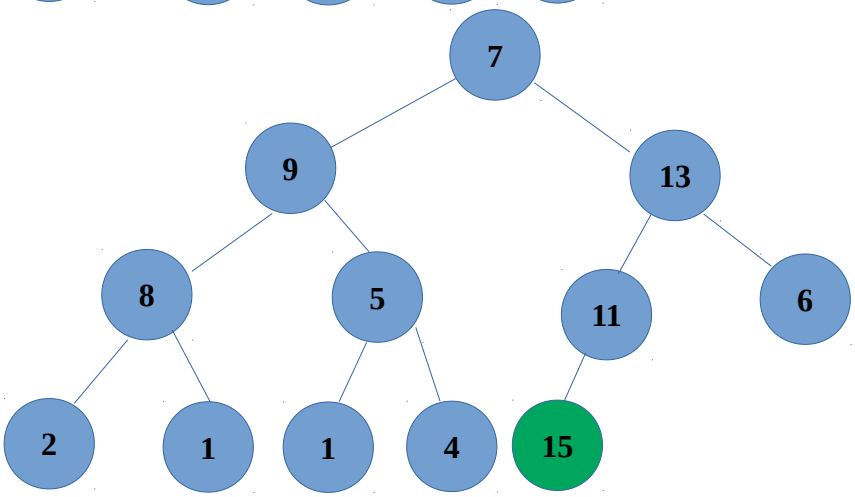

15
2
13
9
1
7
6
8
1
5
4
11

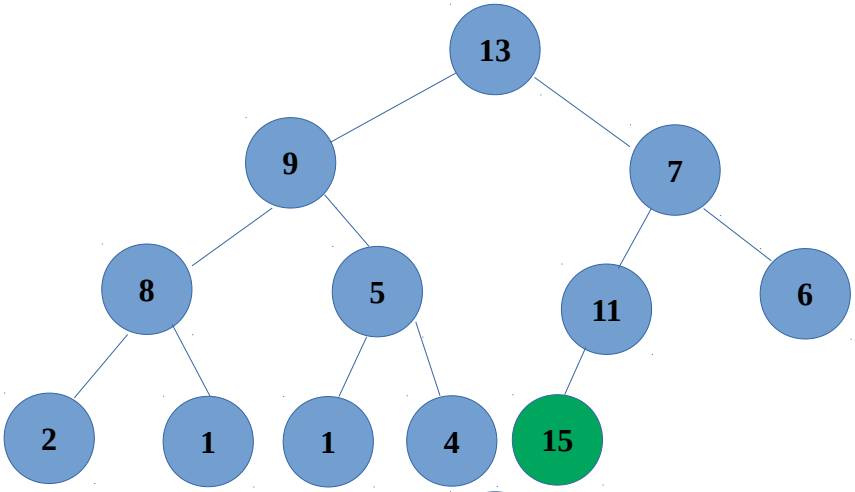

15
9
13
2
1
7
6
8
1
5
4
11

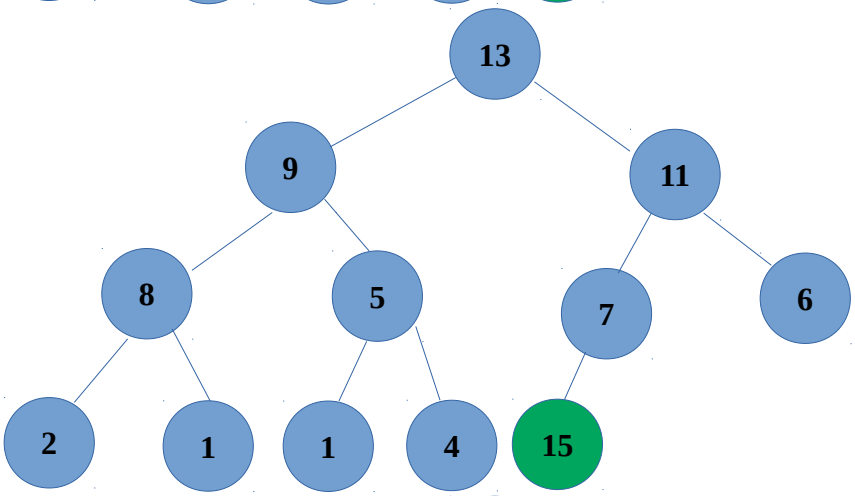

15
9
13
8
1
7
6
2
1
5
4
11

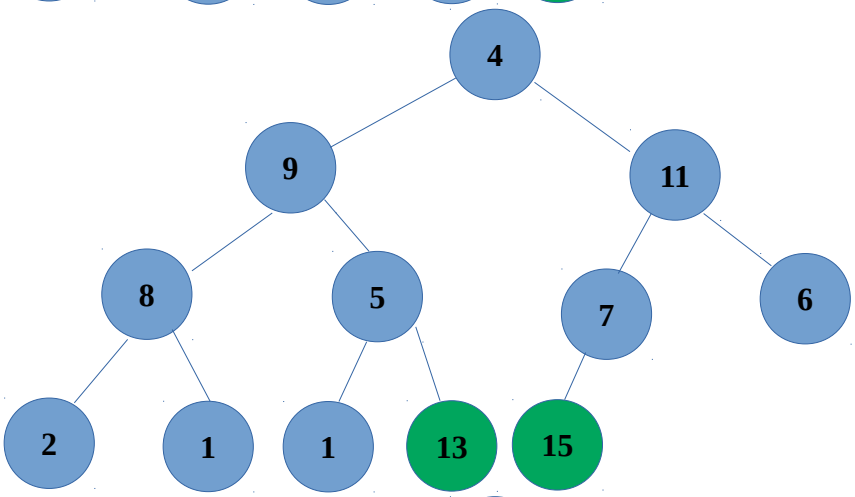

15
9
13
8
5
7
6
2
1
1
4
11

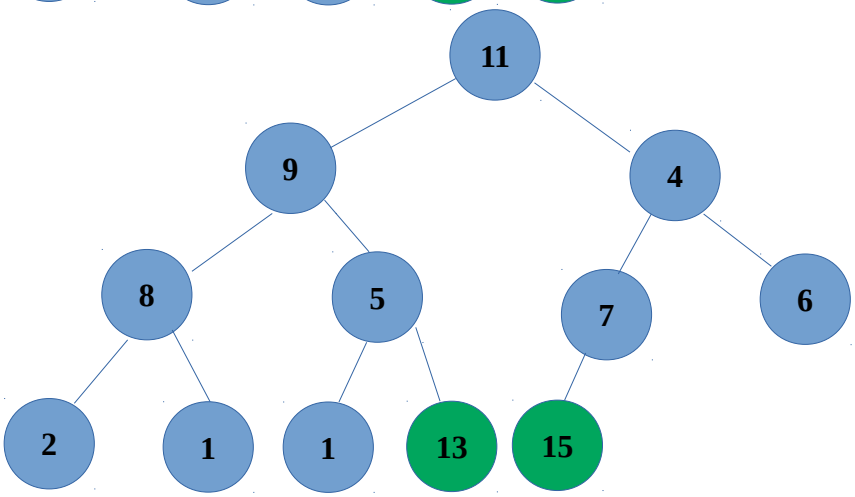

15
9
13
8
5
11
6
2
1
1
4
7

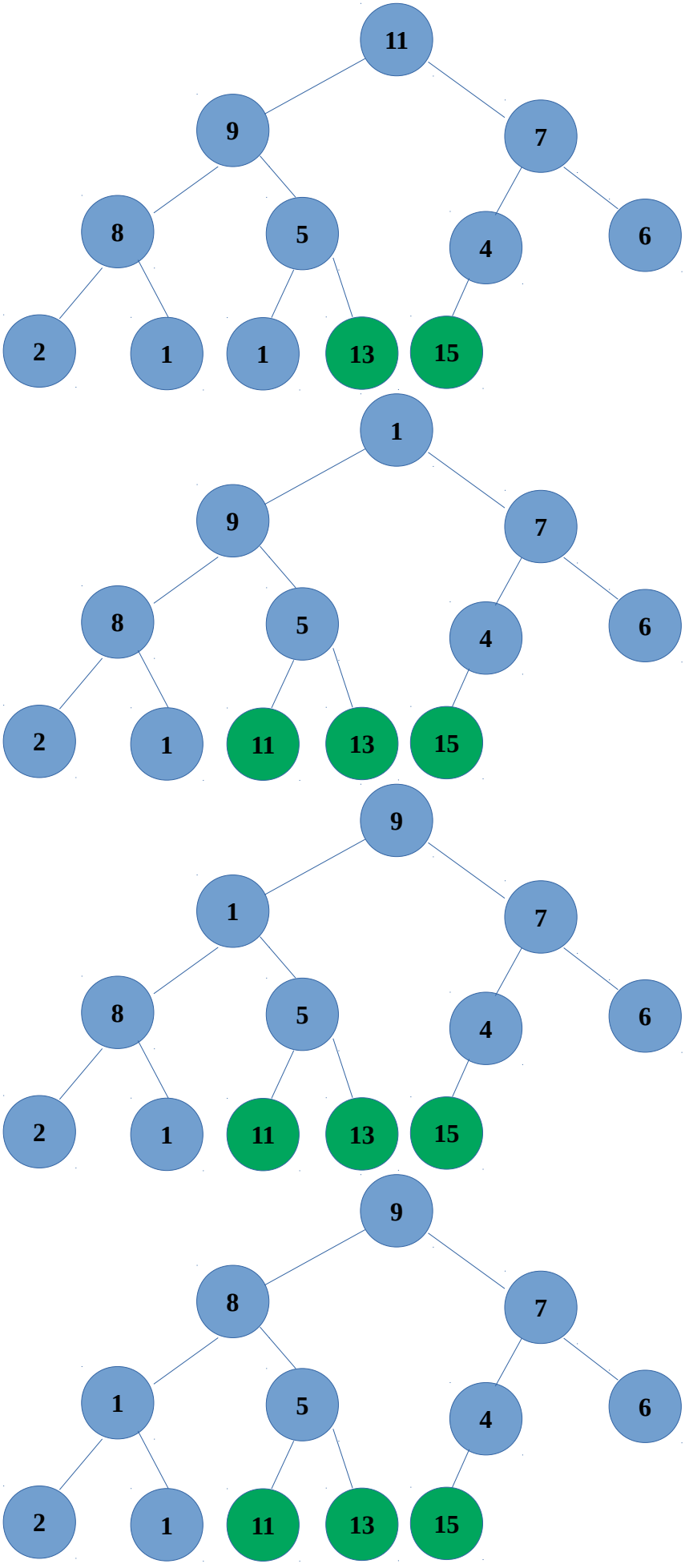

7
9
13
8
5
11
6
2
1
1
4
15

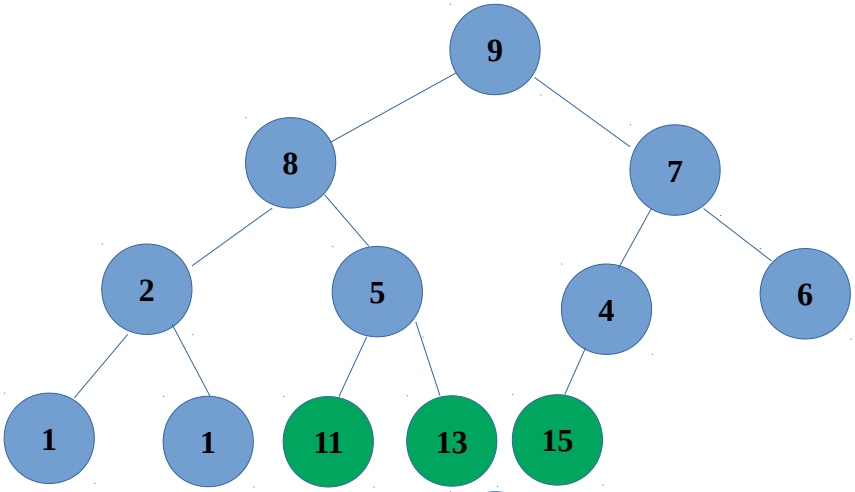

13
9
7
8
5
11
6
2
1
1
4
15

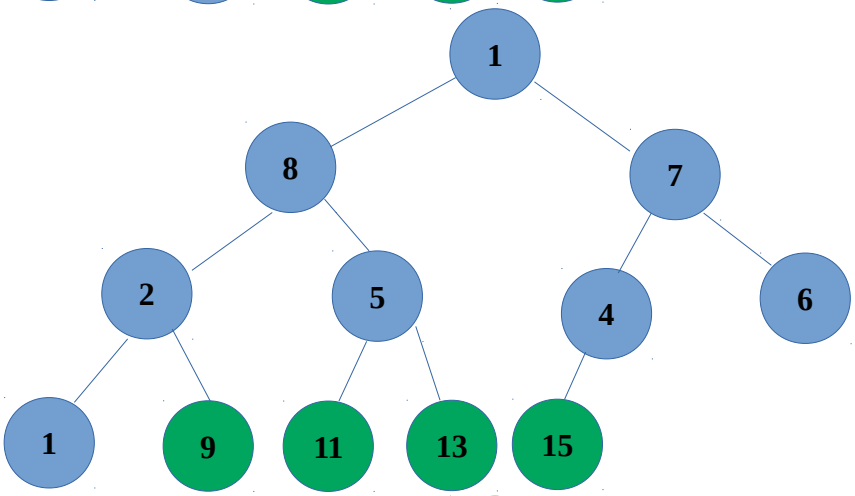

13
9
11
8
5
7
6
2
1
1
4
15

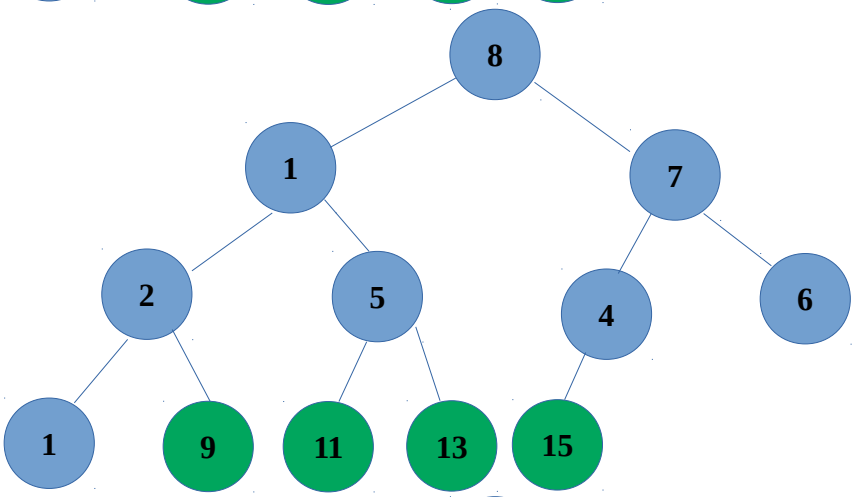

4
9
11
8
5
7
6
2
1
1
13
15

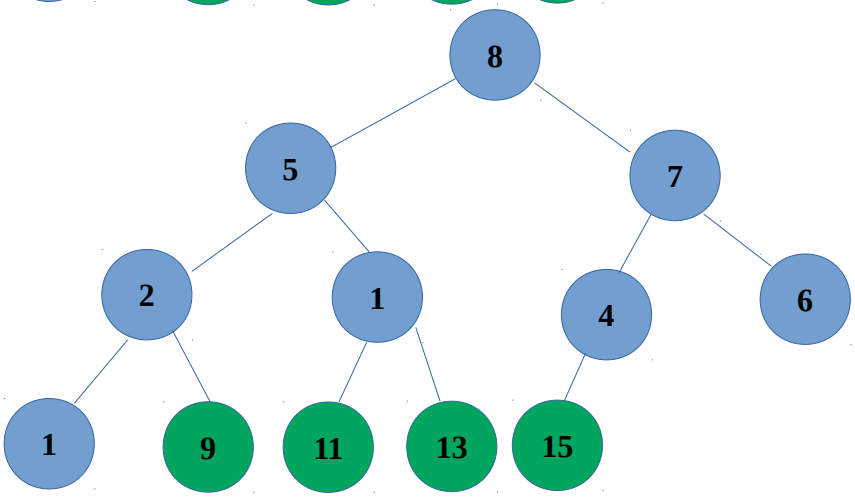

11
9
4
8
5
7
6
2
1
1
13
15

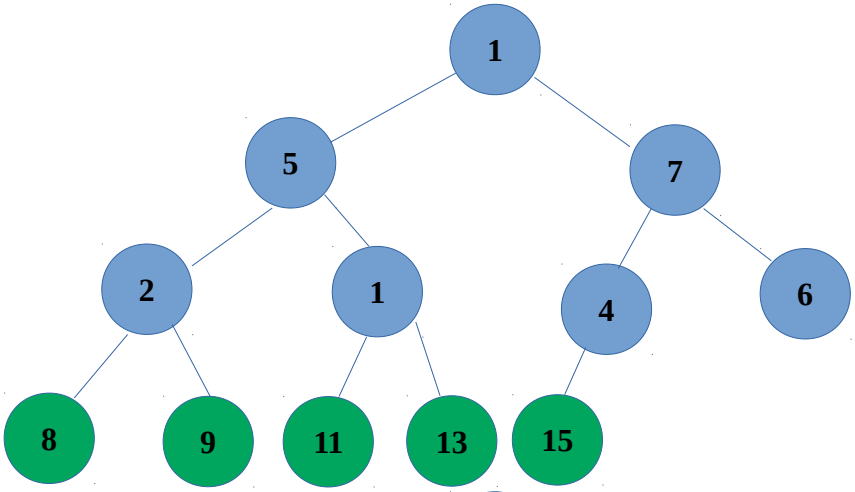

11
9
7
8
5
4
6
2
1
1
13
15
1
9
7
8
5
4
6
2
1
11
13
15
9
1
7
8
5
4
6
2
1
11
13
15
9
8
7
1
5
4
6
2
1
11
13
15

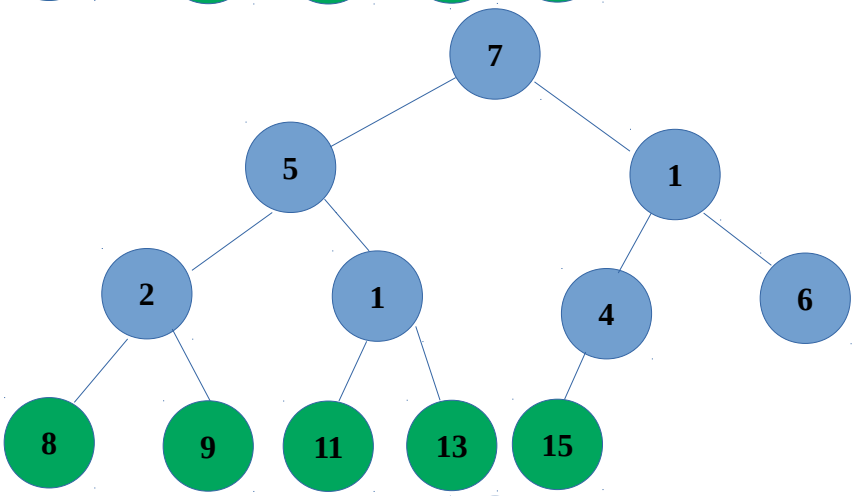

9
8
7
2
5
4
6
1
1
11
13
15

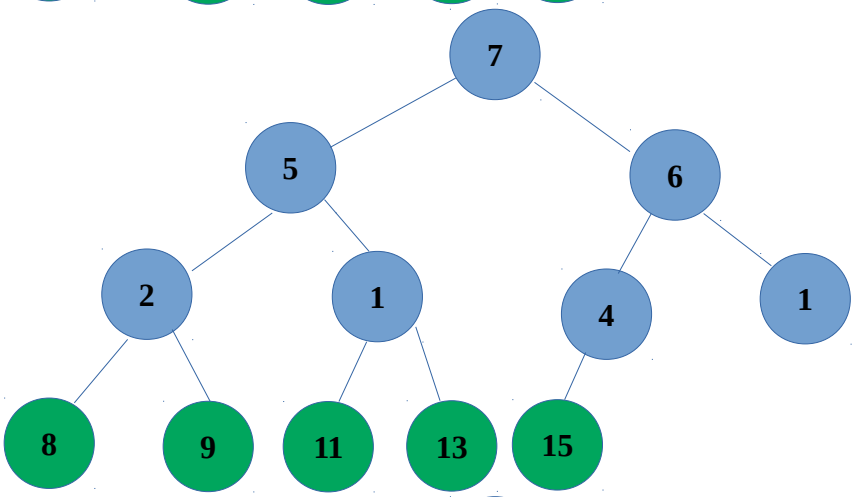

1
8
7
2
5
4
6
1
9
11
13
15

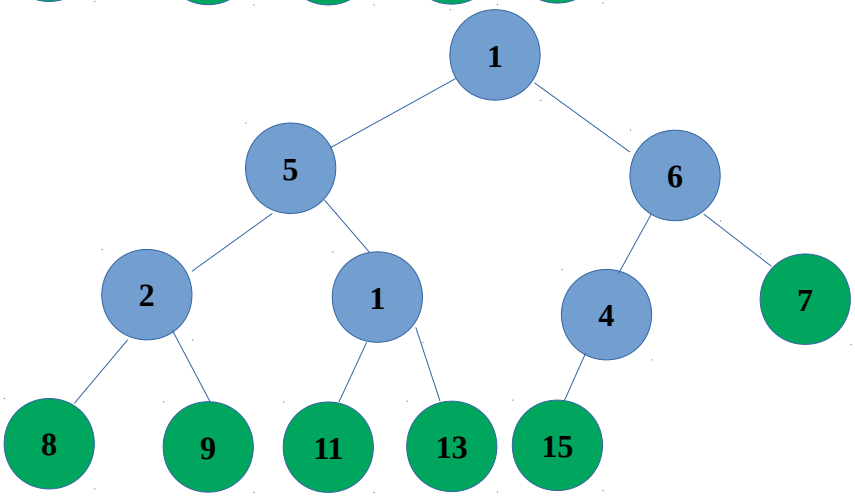

8
1
7
2
5
4
6
1
9
11
13
15

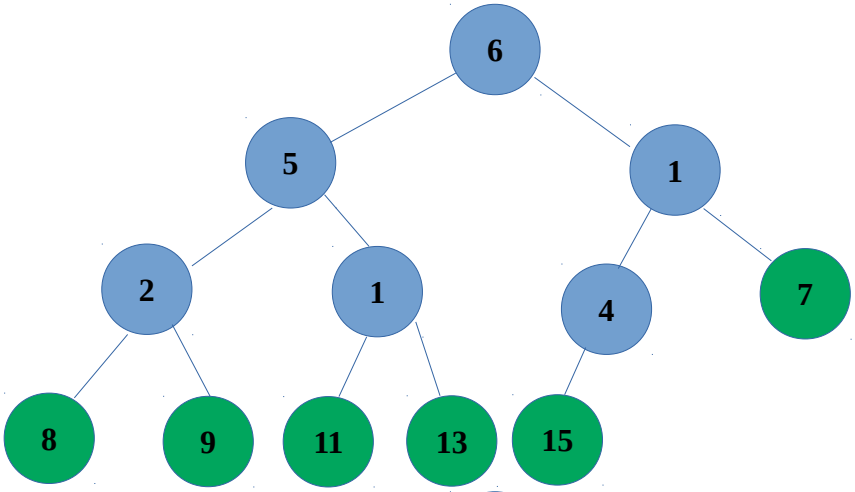

8
5
7
2
1
4
6
1
9
11
13
15

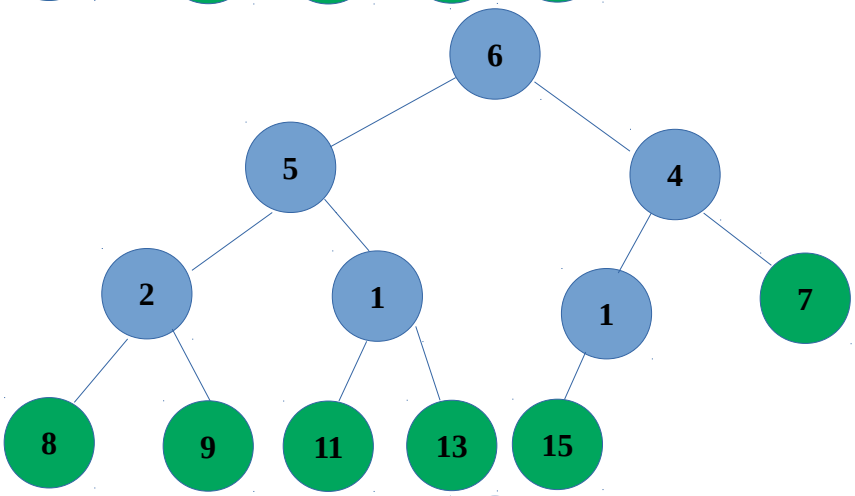

1
5
7
2
1
4
6
8
9
11
13
15

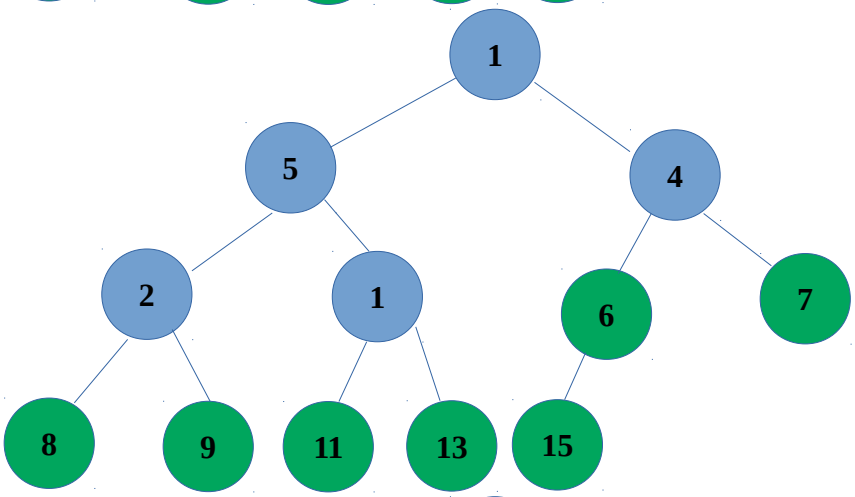

7
5
1
2
1
4
6
8
9
11
13
15

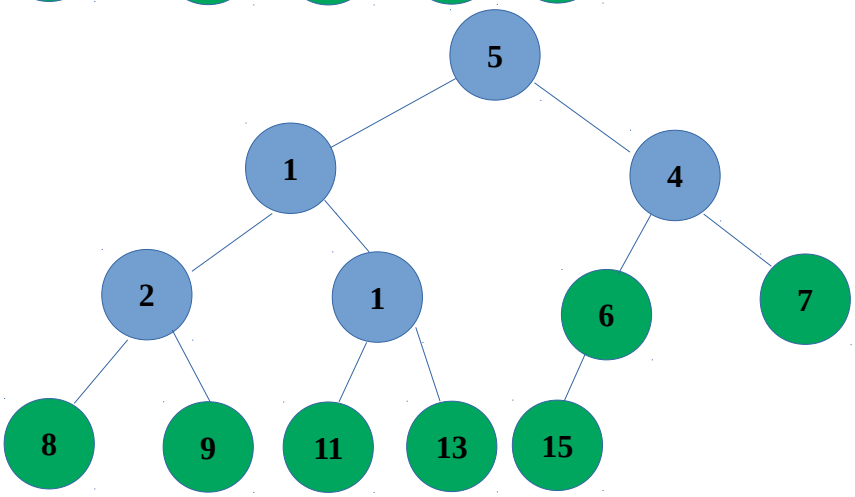

7
5
6
2
1
4
1
8
9
11
13
15

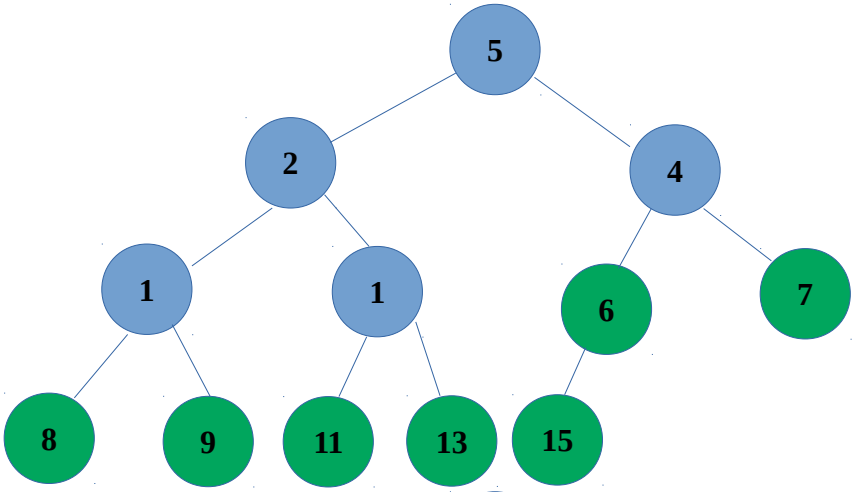

1
5
6
2
1
4
7
8
9
11
13
15

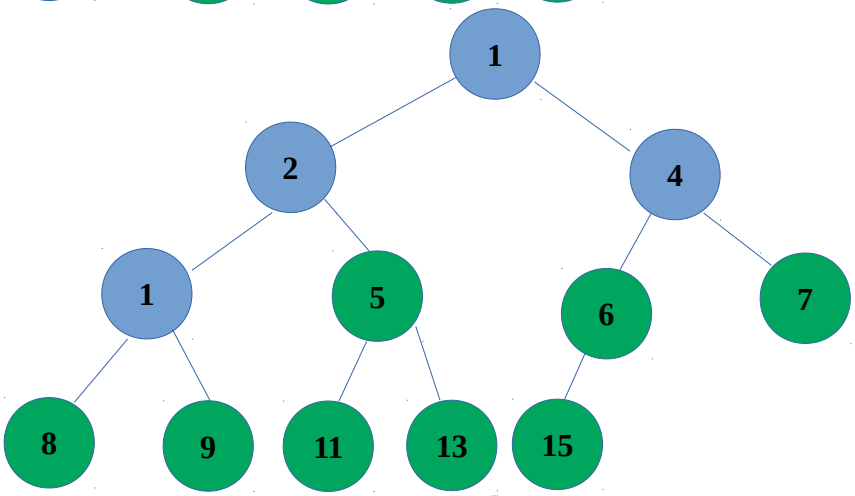

6
5
1
2
1
4
7
8
9
11
13
15

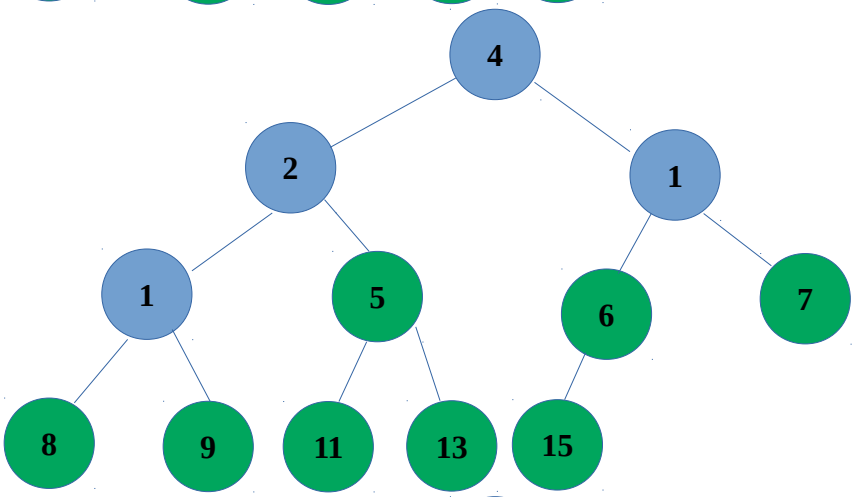

6
5
4
2
1
1
7
8
9
11
13
15

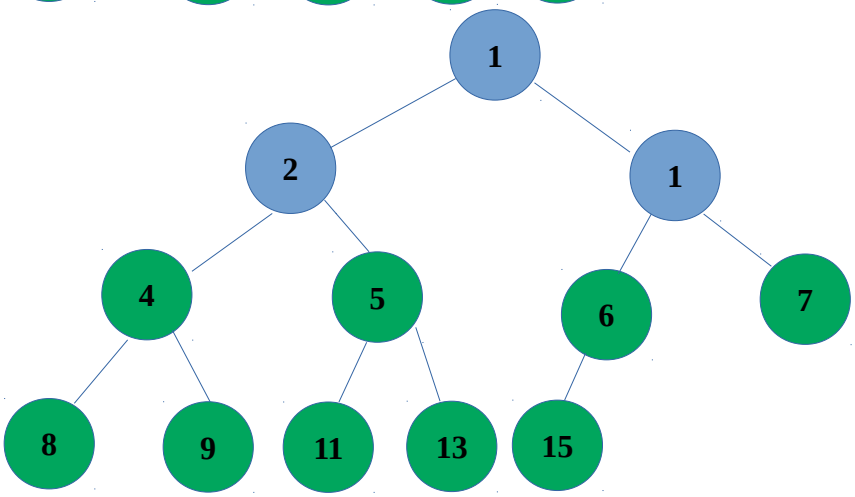

1
5
4
2
1
6
7
8
9
11
13
15

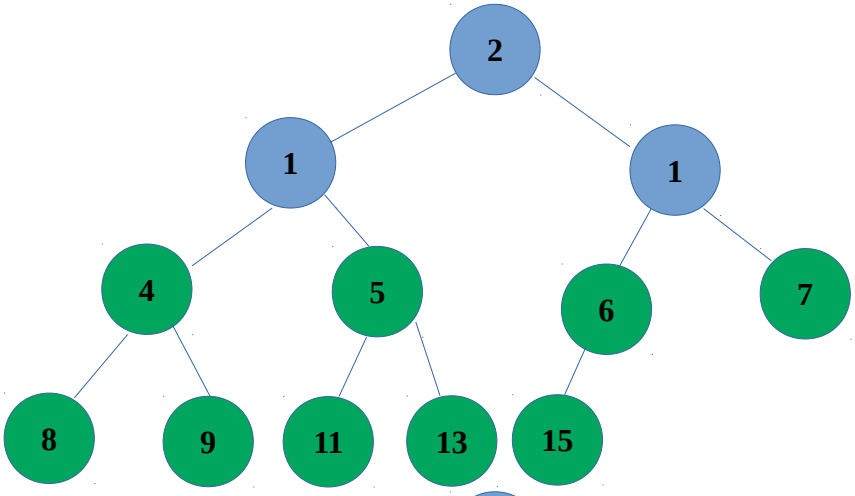

5
1
4
2
1
6
7
8
9
11
13
15

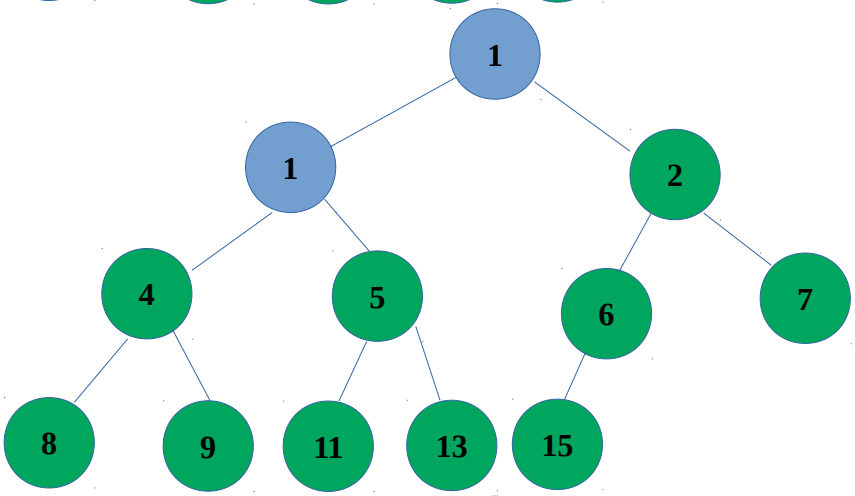

5
2
4
1
1
6
7
8
9
11
13
15

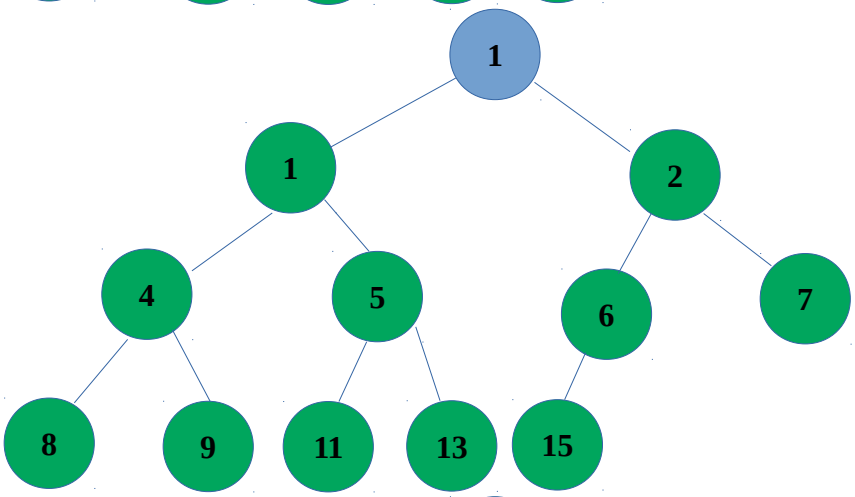

1
2
4
1
5
6
7
8
9
11
13
15

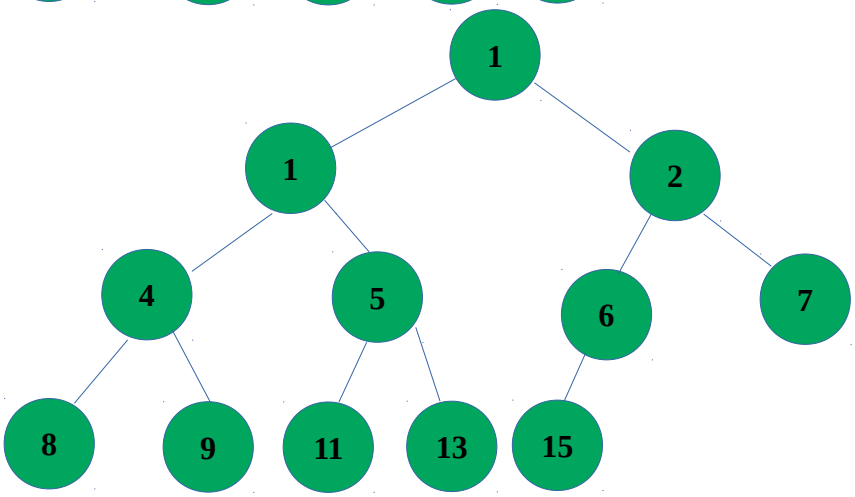

4
2
1
1
5
6
7
8
9
11
13
15


1
2
1
4
5
6
7
8
9
11
13
15


2
1
1
4
5
6
7
8
9
11
13
15

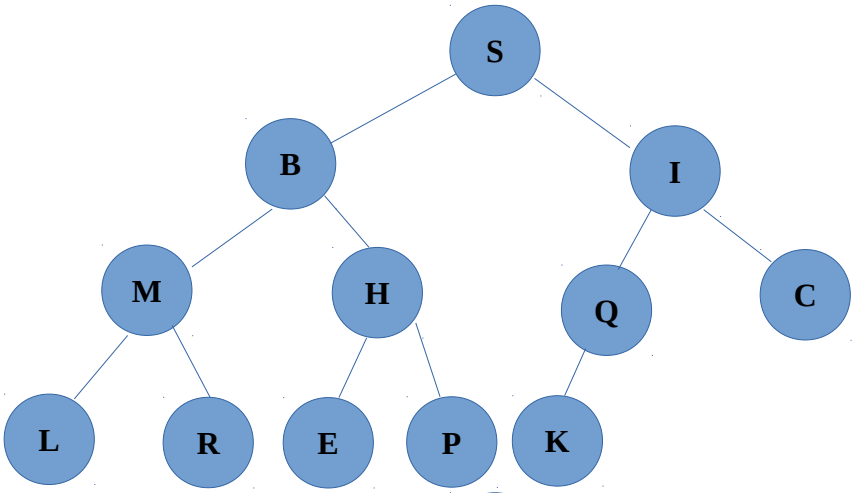

1
1
2
4
5
6
7
8
9
11
13
15

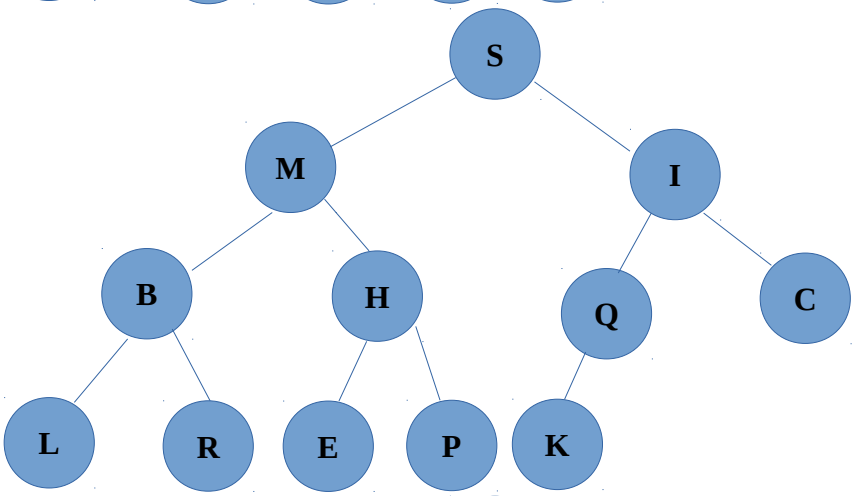

1
1
2
4
5
6
7
8
9
11
13
15

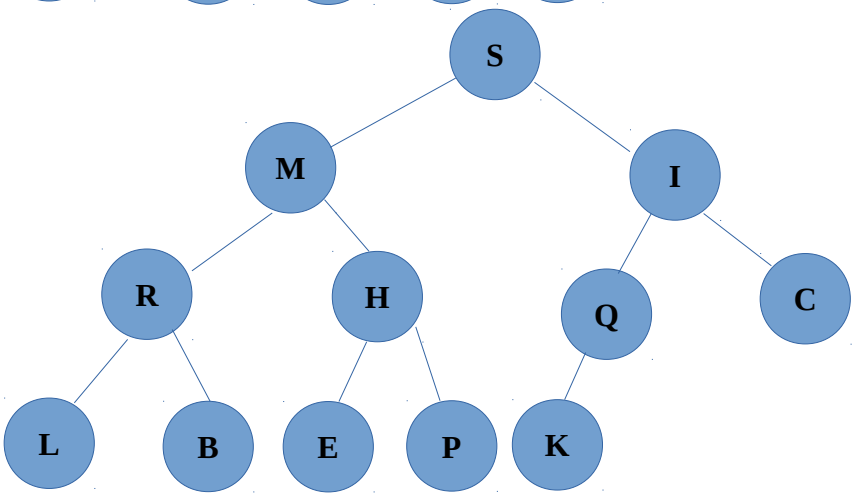

1
1
2
4
5
6
7
8
9
11
13
15

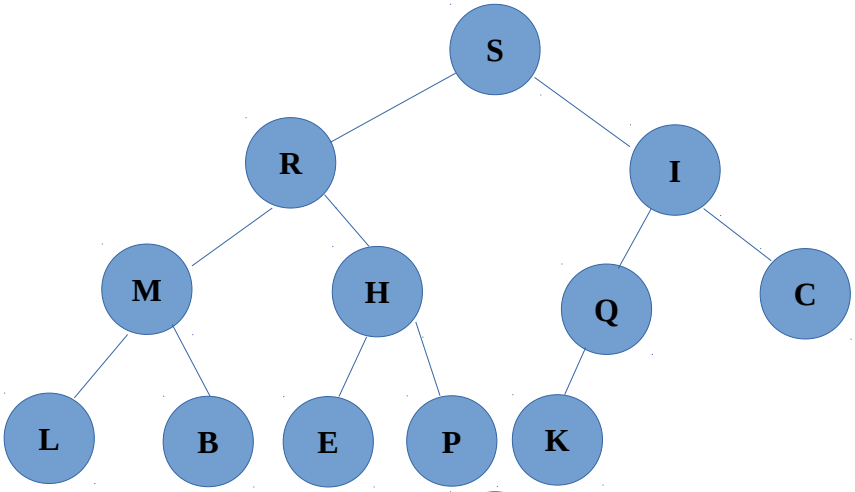
*D={‘S’,’B’,’I’,’M’,’H’,’Q’,’C’,’L’,’R’,’E’,’P’,’K’}

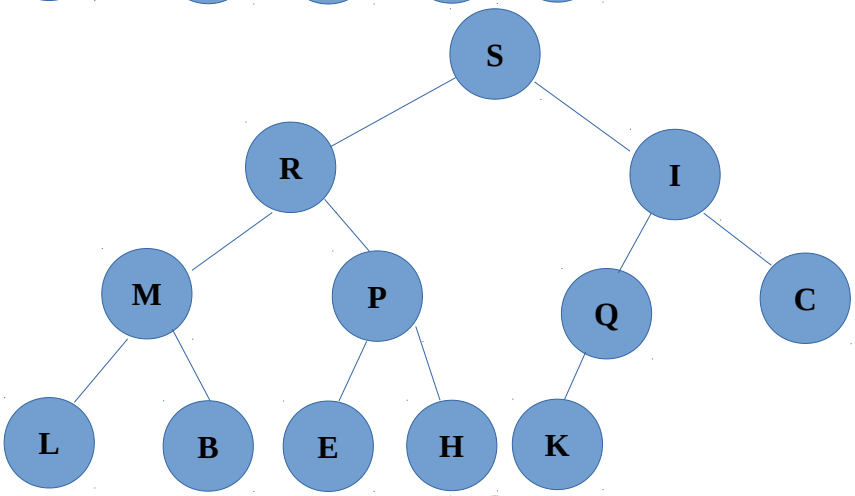
Construct heap following list of unsorted numbers

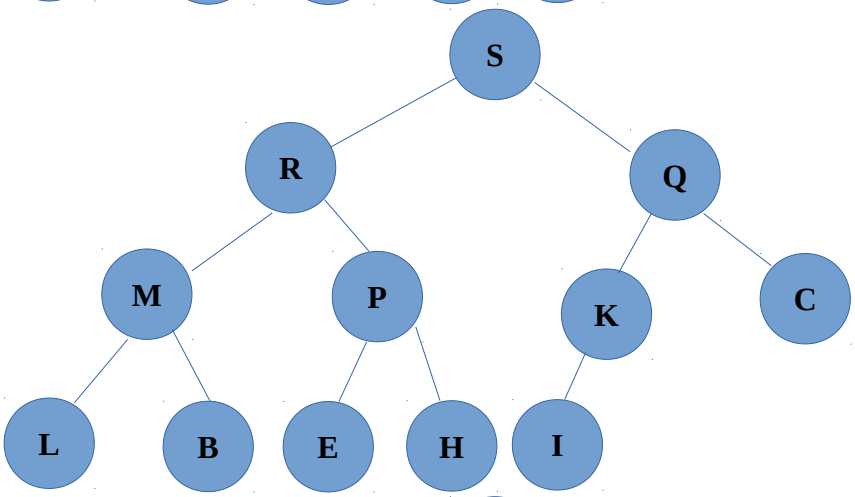


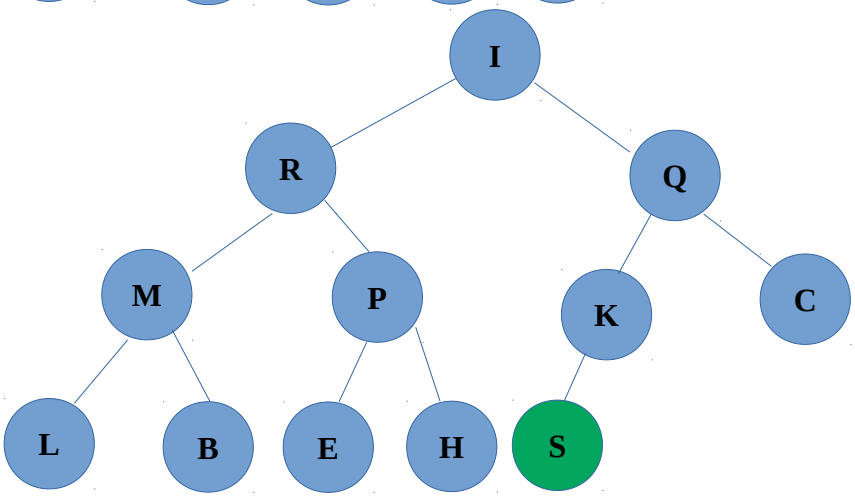


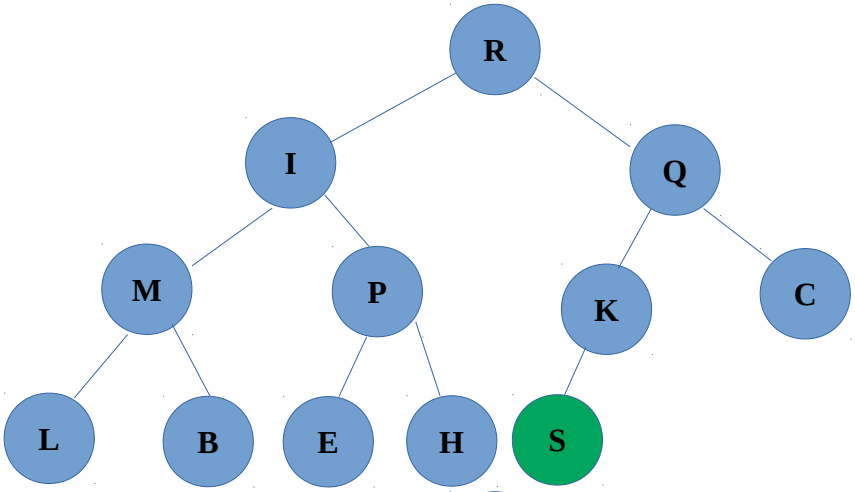


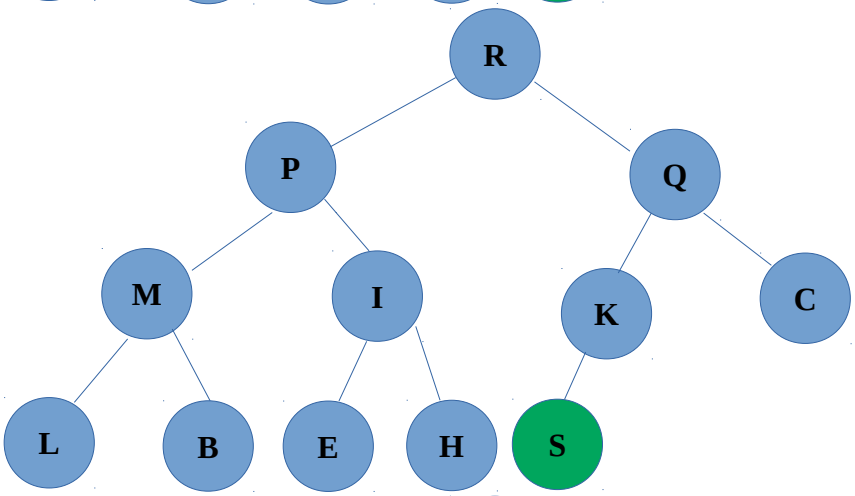

S
R
I
M
H
Q
C
L
B
E
P
K

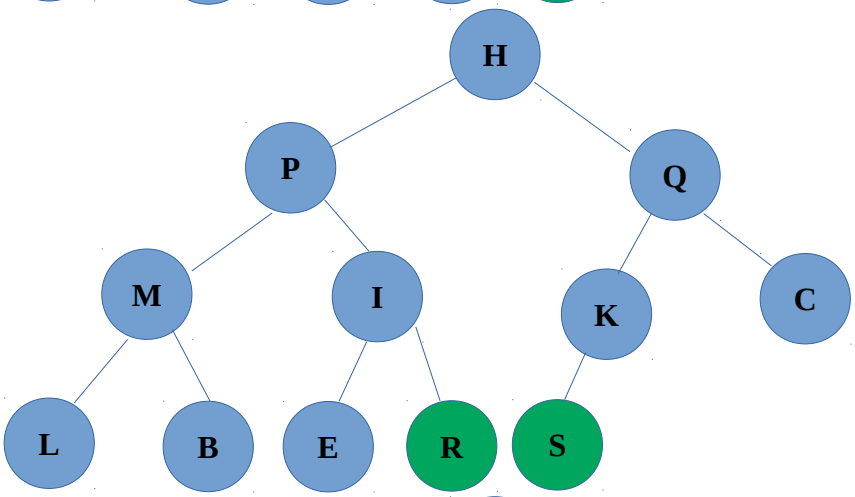

S
R
I
M
P
Q
C
L
B
E
H
K

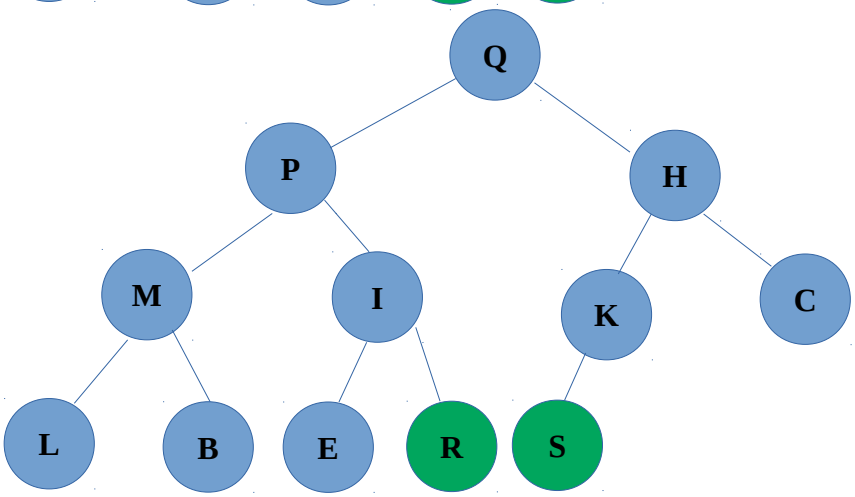

S
R
Q
M
P
K
C
L
B
E
H
I

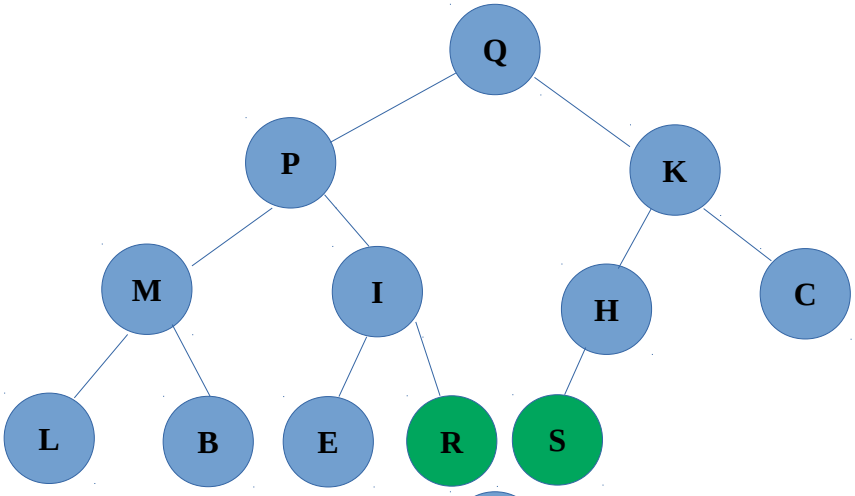

I
R
Q
M
P
K
C
L
B
E
H
S

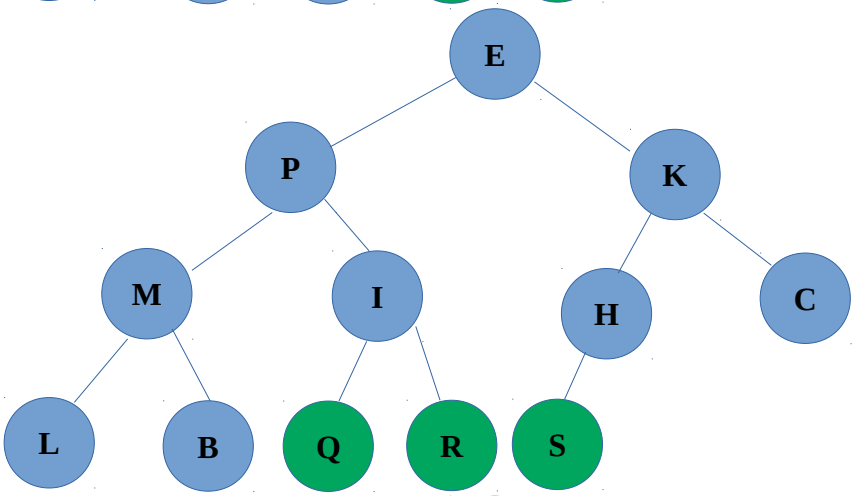

R
I
Q
M
P
K
C
L
B
E
H
S

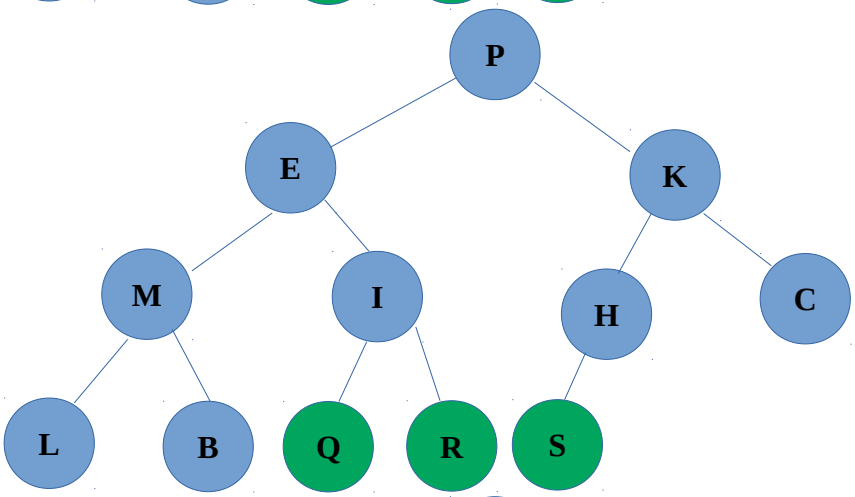

R
P
Q
M
I
K
C
L
B
E
H
S

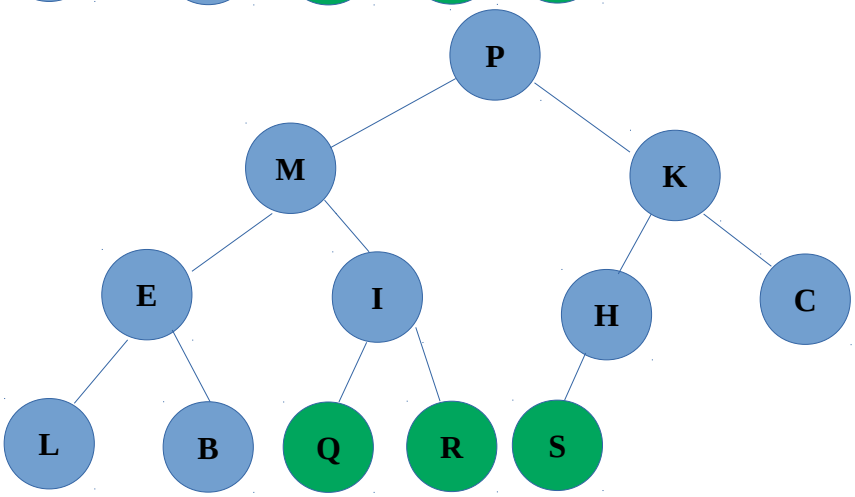

H
P
Q
M
I
K
C
L
B
E
R
S

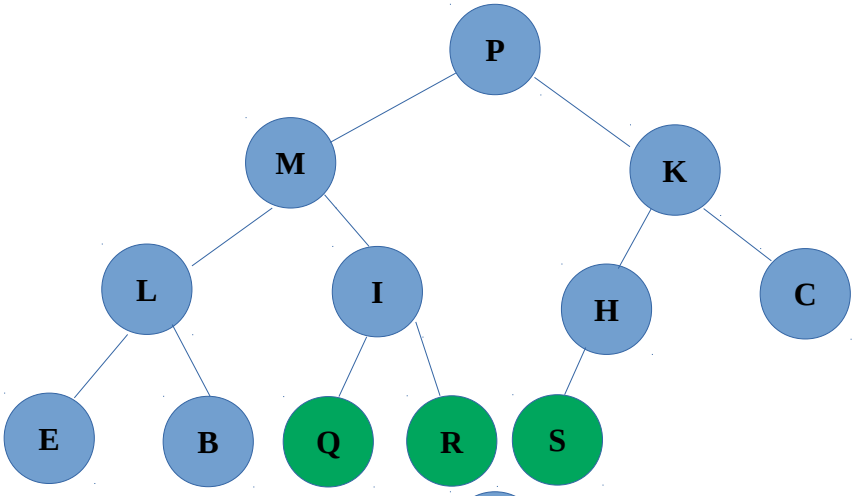

Q
P
H
M
I
K
C
L
B
E
R
S

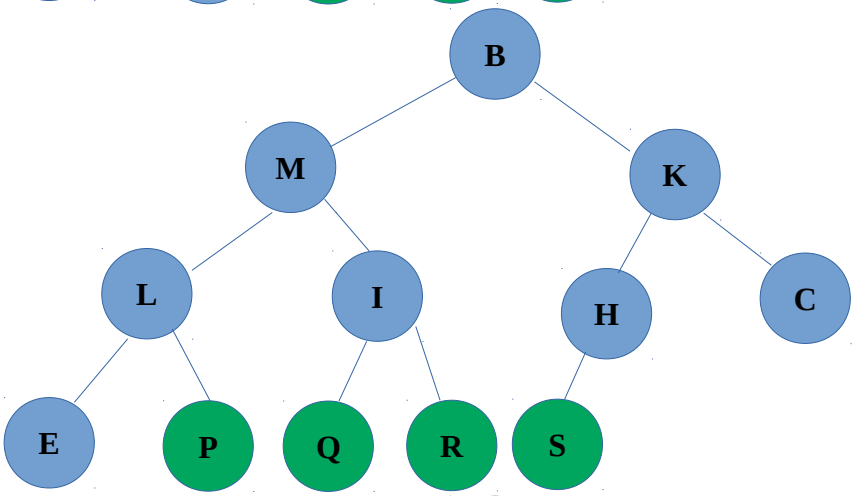

Q
P
K
M
I
H
C
L
B
E
R
S

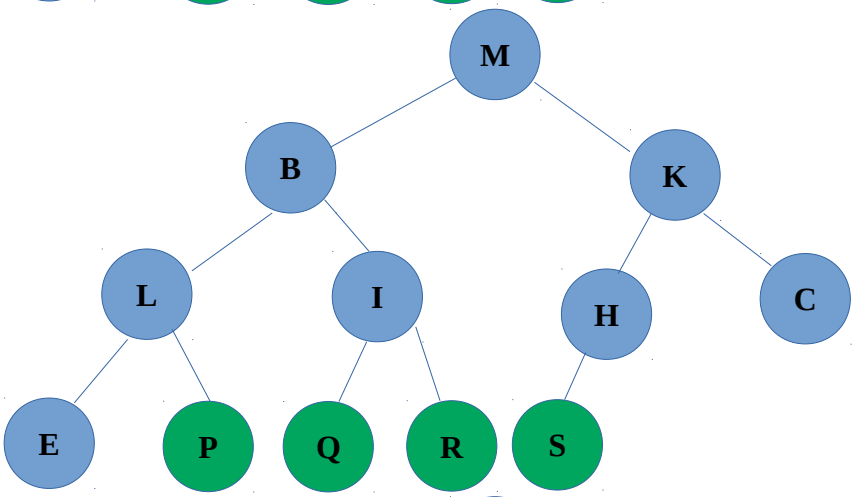

E
P
K
M
I
H
C
L
B
Q
R
S

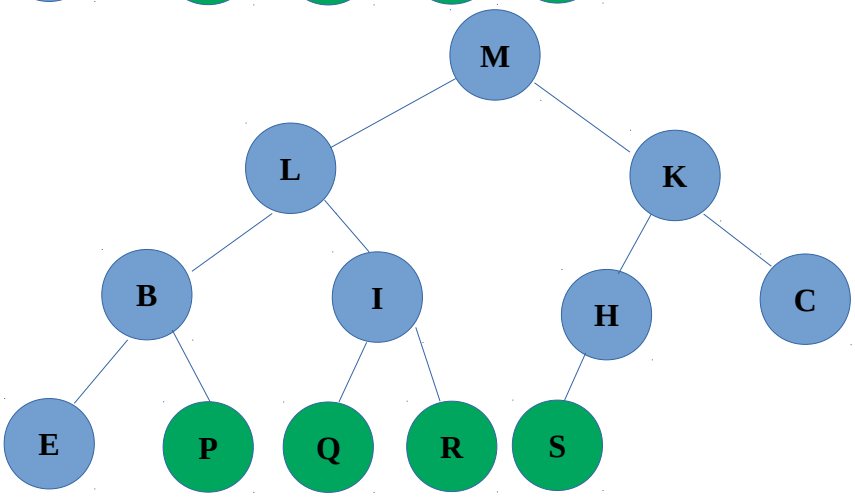

P
E
K
M
I
H
C
L
B
Q
R
S

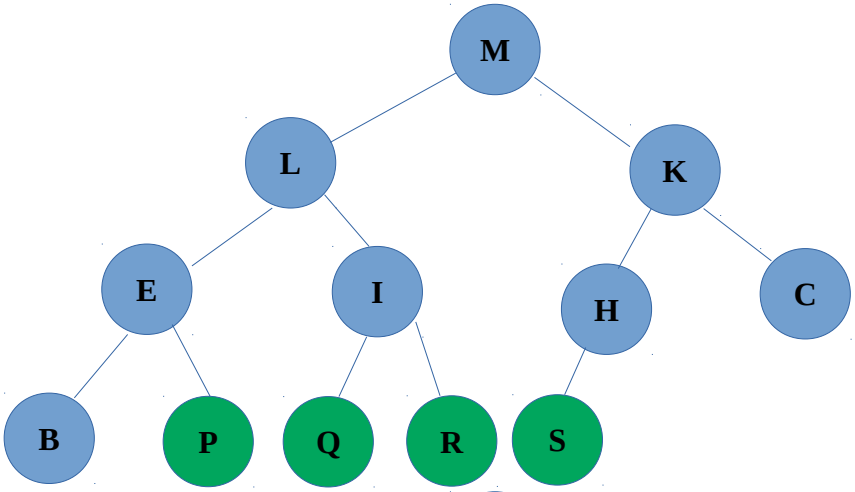

P
M
K
E
I
H
C
L
B
Q
R
S

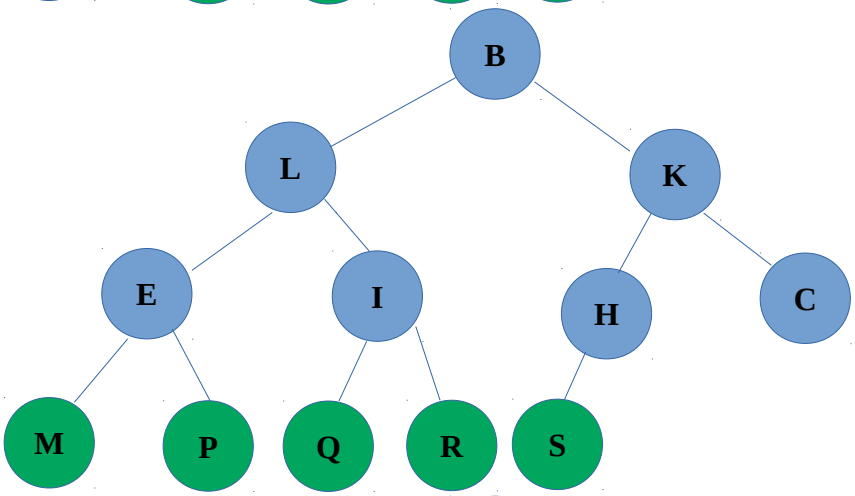

P
M
K
L
I
H
C
E
B
Q
R
S

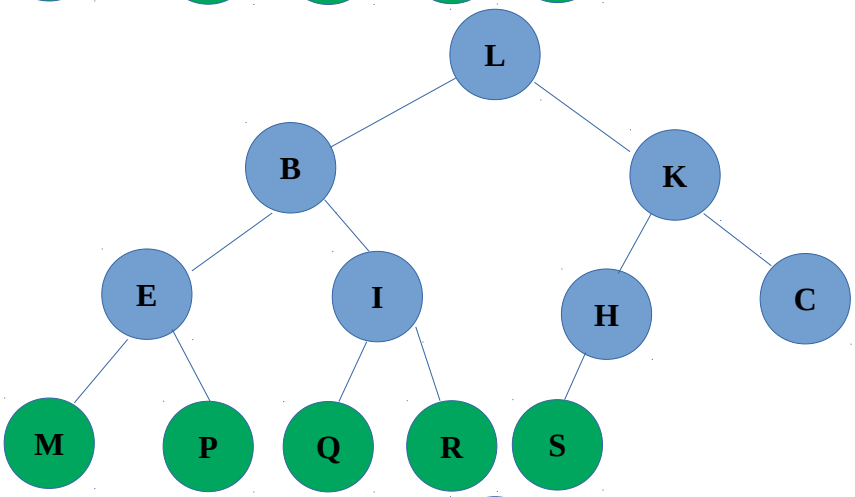

B
M
K
L
I
H
C
E
P
Q
R
S

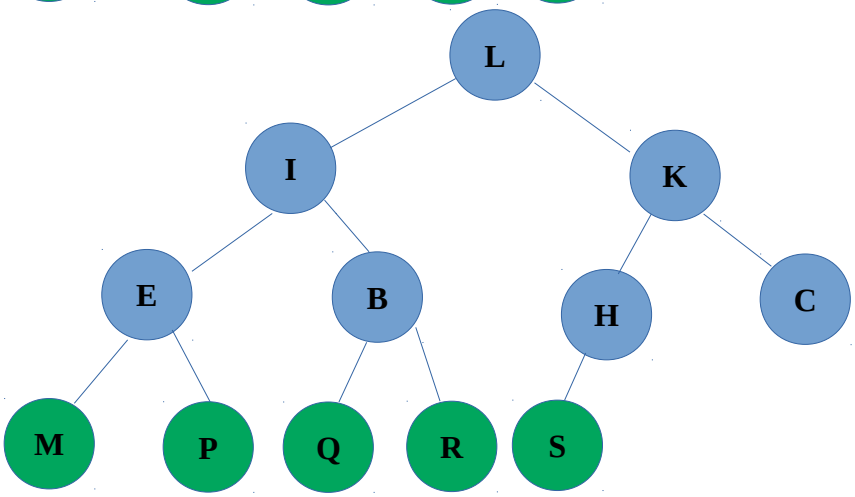

M
B
K
L
I
H
C
E
P
Q
R
S

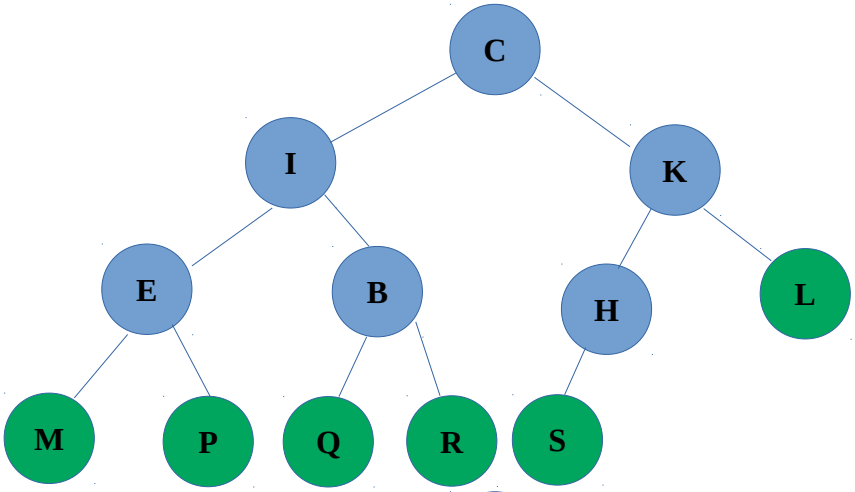

M
L
K
B
I
H
C
E
P
Q
R
S

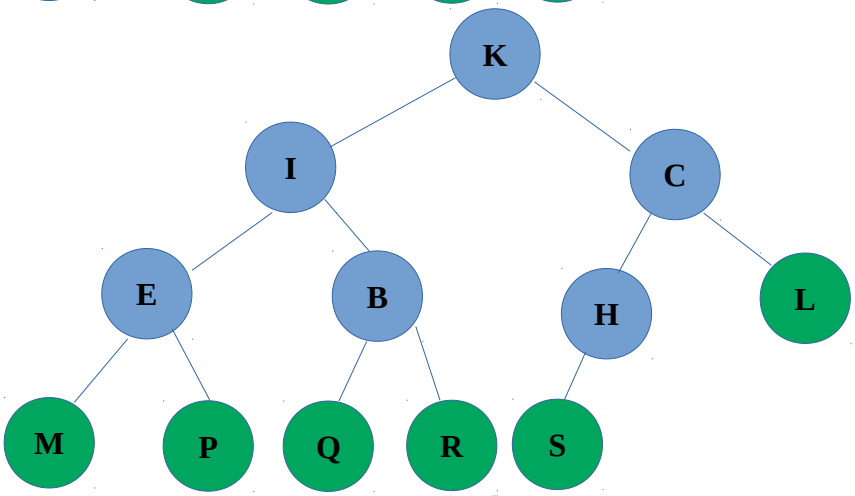

M
L
K
E
I
H
C
B
P
Q
R
S

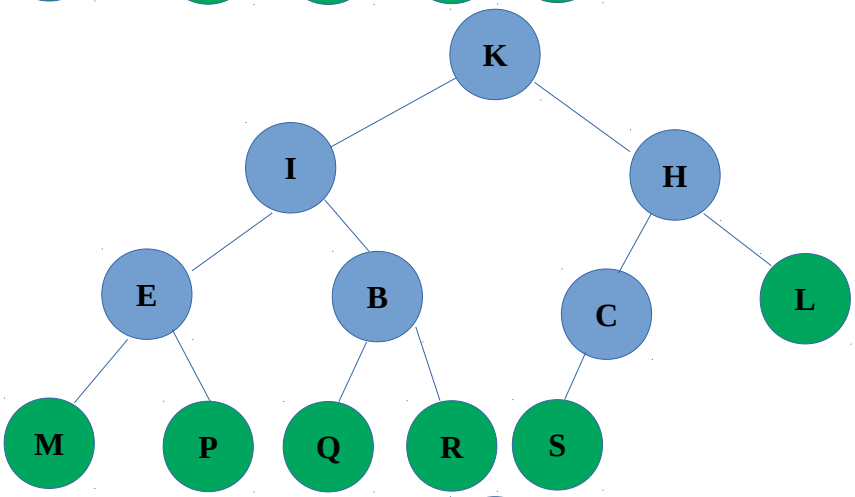

B
L
K
E
I
H
C
M
P
Q
R
S

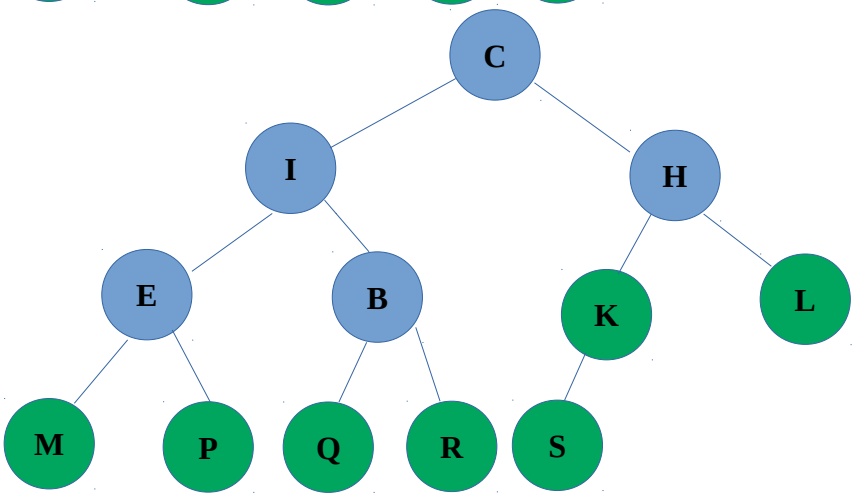

L
B
K
E
I
H
C
M
P
Q
R
S

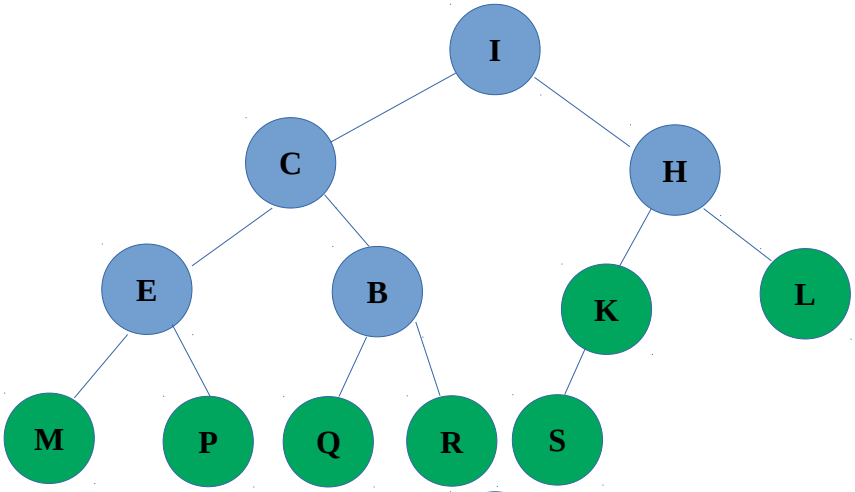

L
I
K
E
B
H
C
M
P
Q
R
S

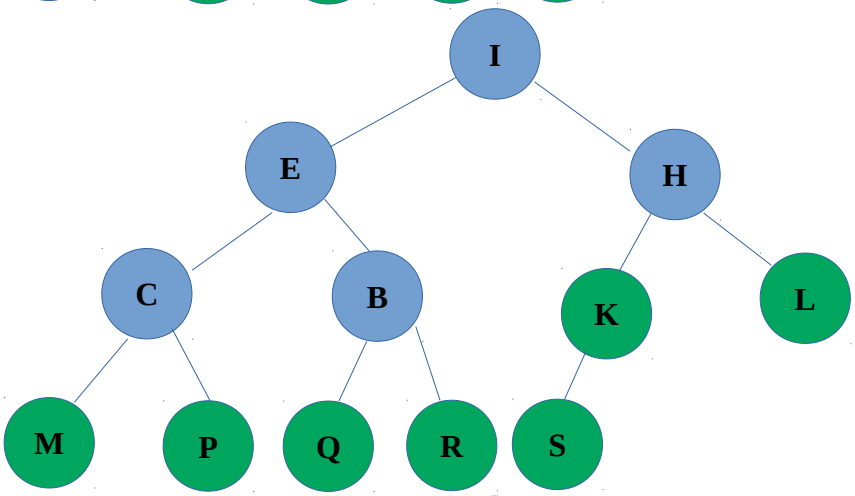

C
I
K
E
B
H
L
M
P
Q
R
S

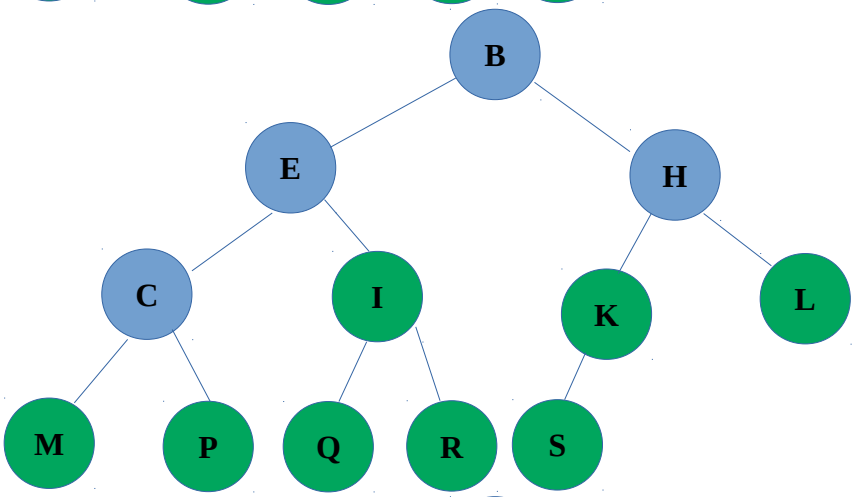

K
I
C
E
B
H
L
M
P
Q
R
S

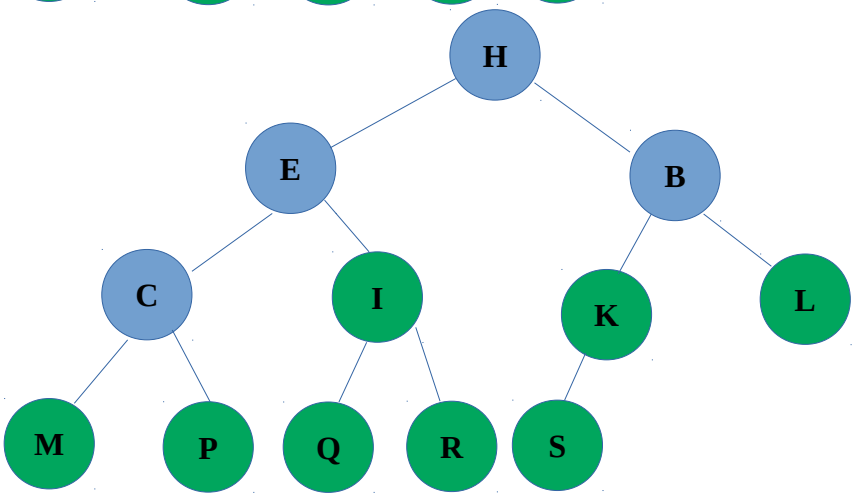

K
I
H
E
B
C
L
M
P
Q
R
S

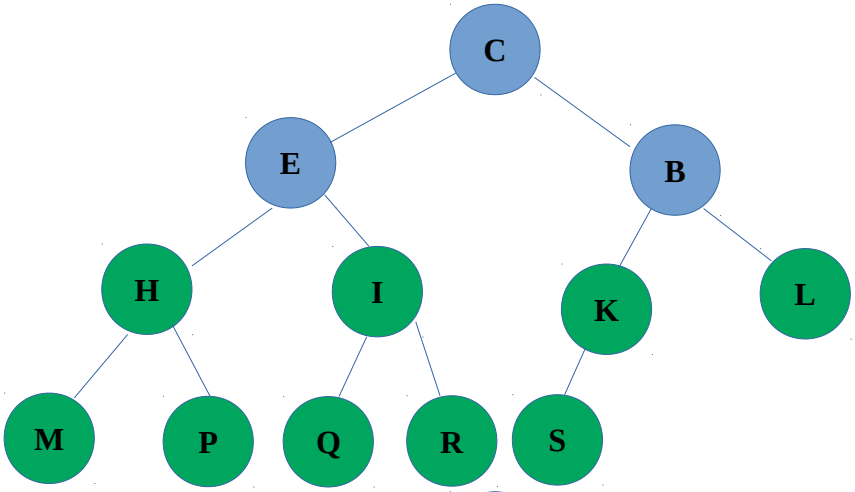

C
I
H
E
B
K
L
M
P
Q
R
S

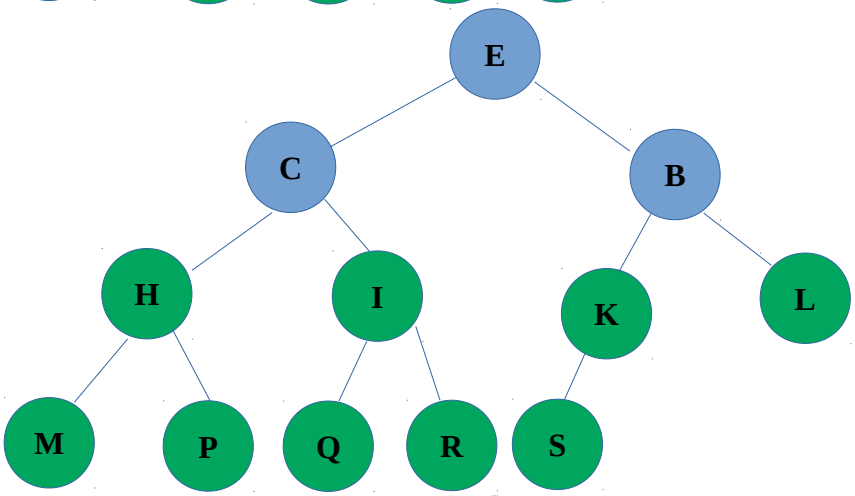

I
C
H
E
B
K
L
M
P
Q
R
S

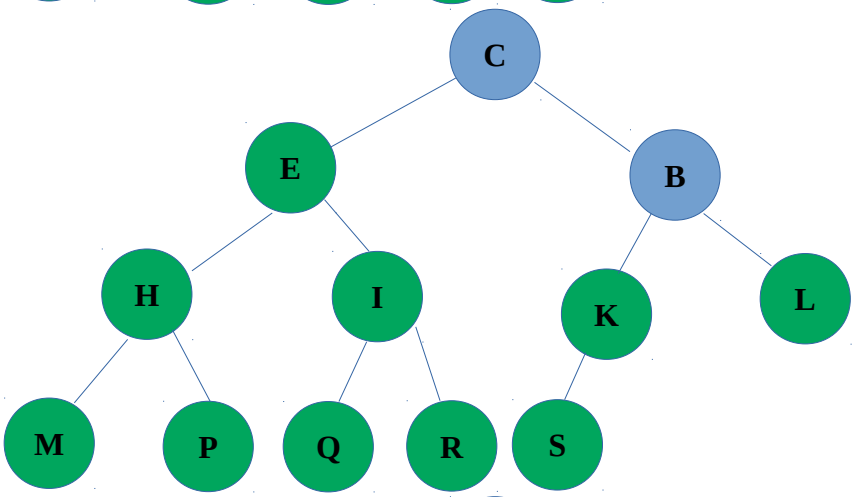

I
E
H
C
B
K
L
M
P
Q
R
S

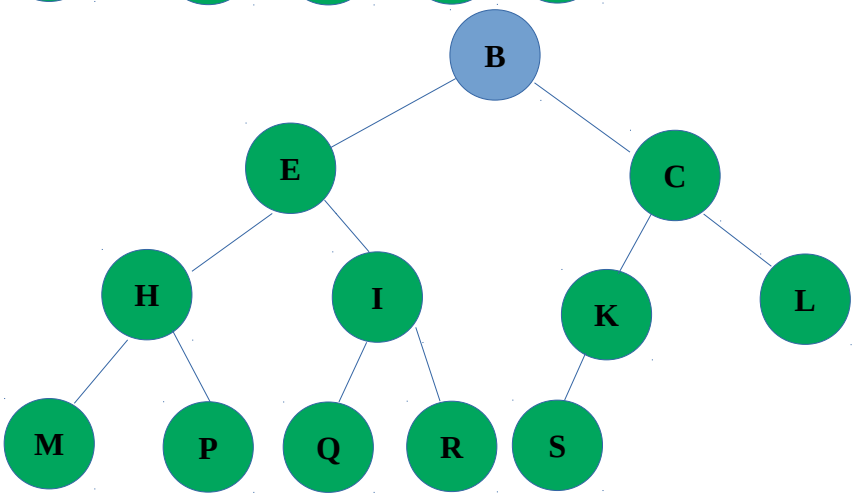

B
E
H
C
I
K
L
M
P
Q
R
S

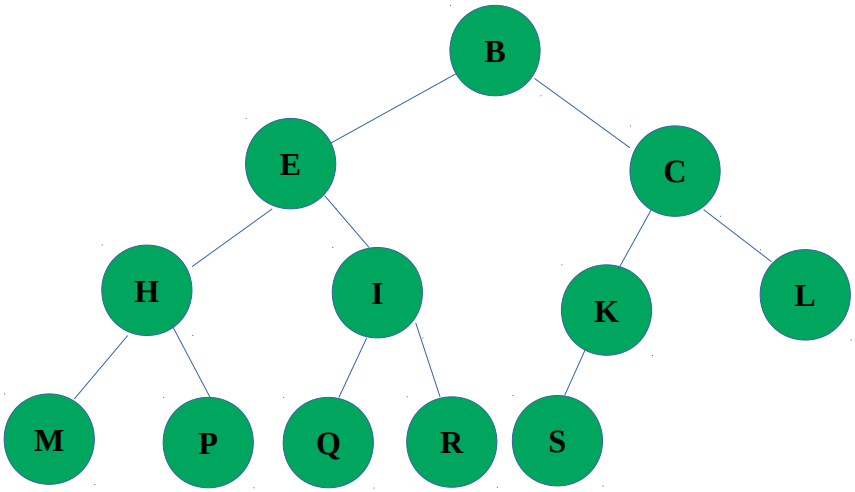

H
E
B
C
I
K
L
M
P
Q
R
S


C
E
B
H
I
K
L
M
P
Q
R
S


E
C
B
H
I
K
L
M
P
Q
R
S

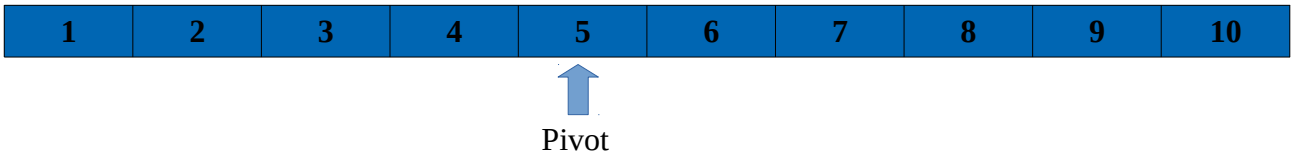

C
E
B
H
I
K
L
M
P
Q
R
S


B
E
C
H
I
K
L
M
P
Q
R
S



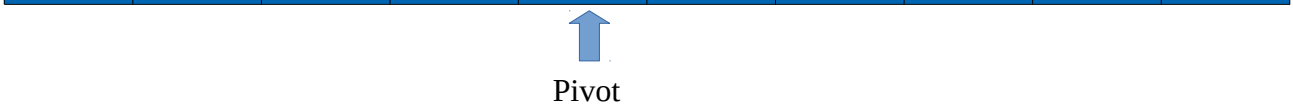
#Quick Sort#

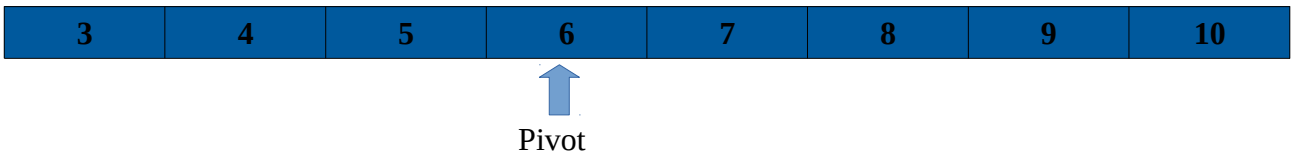
*A={1,2,3,4,5,6,7,8,9,10}

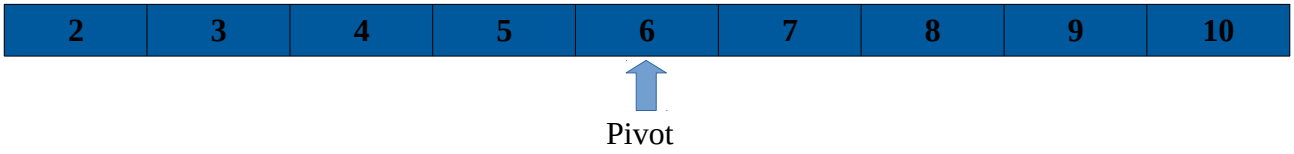


Pivot

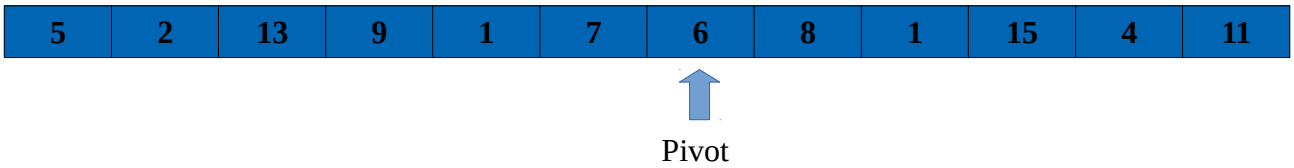
*B={10,9,8,7,6,5,4,3,2,1}

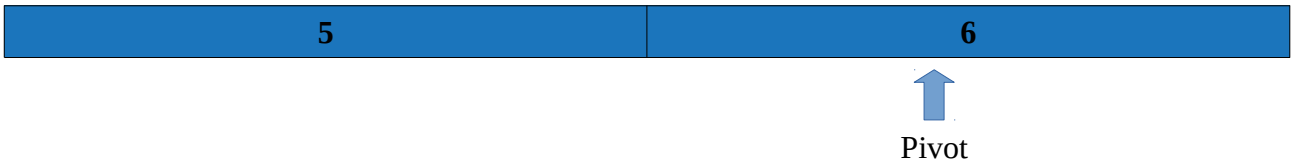


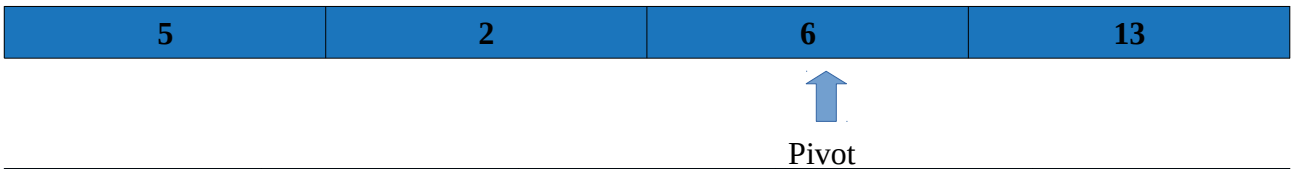


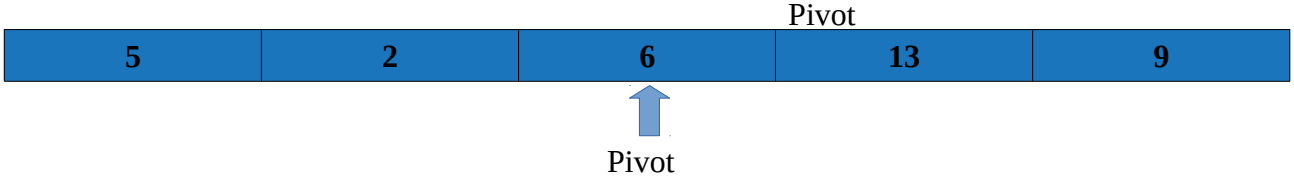


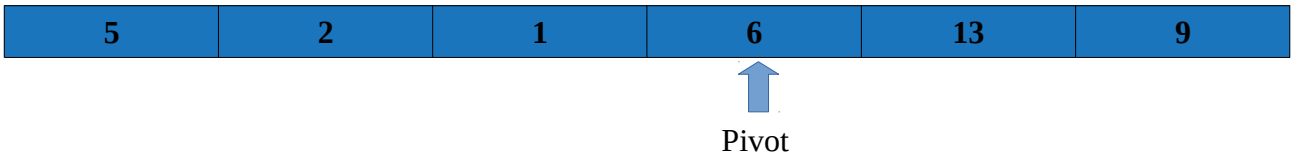
*C={5,2,13,9,1,7,6,8,1,15,4,11}

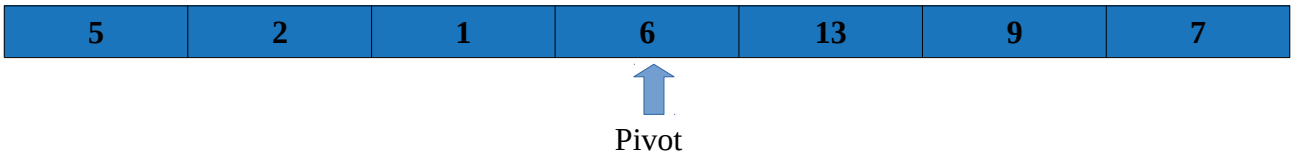


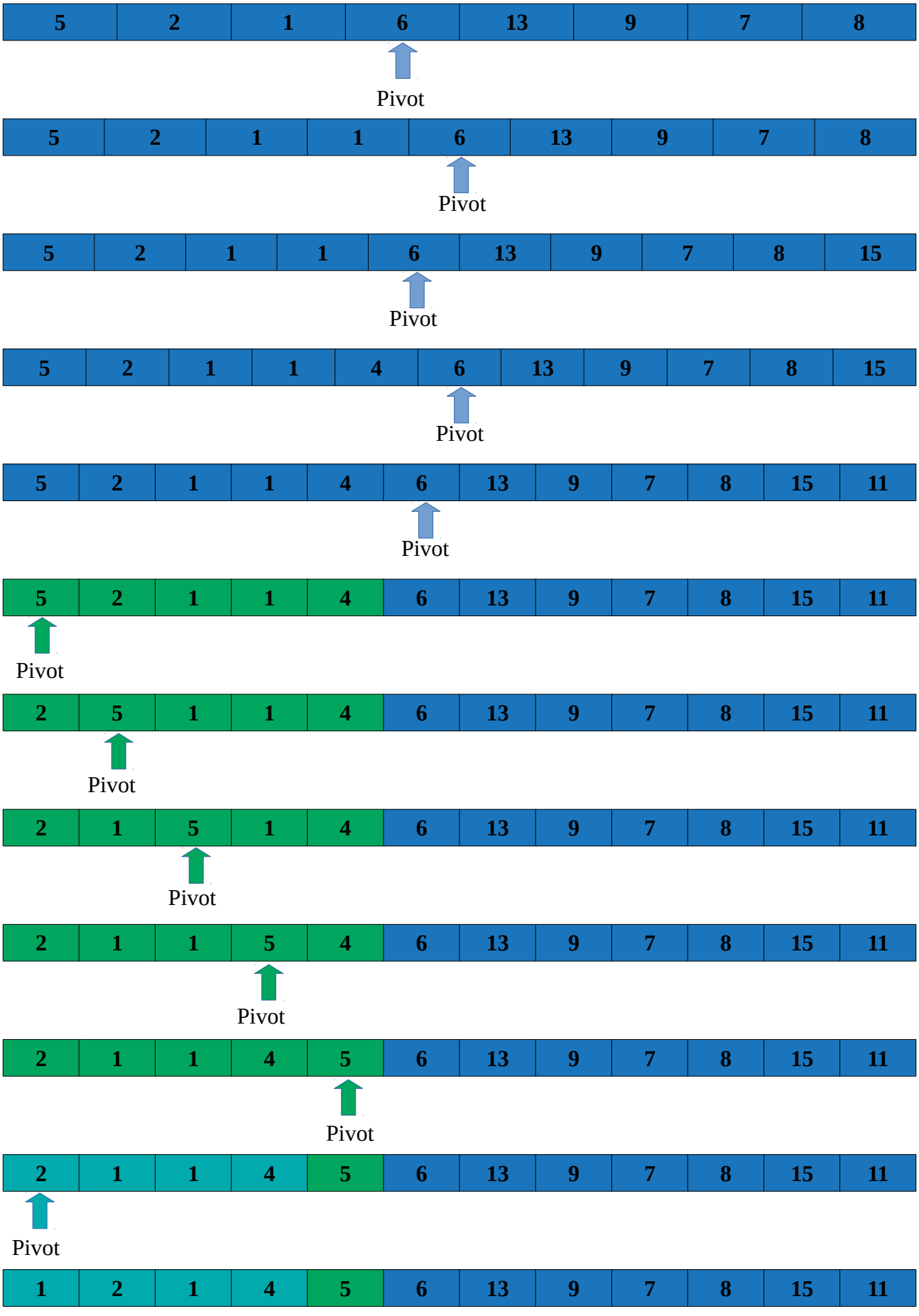


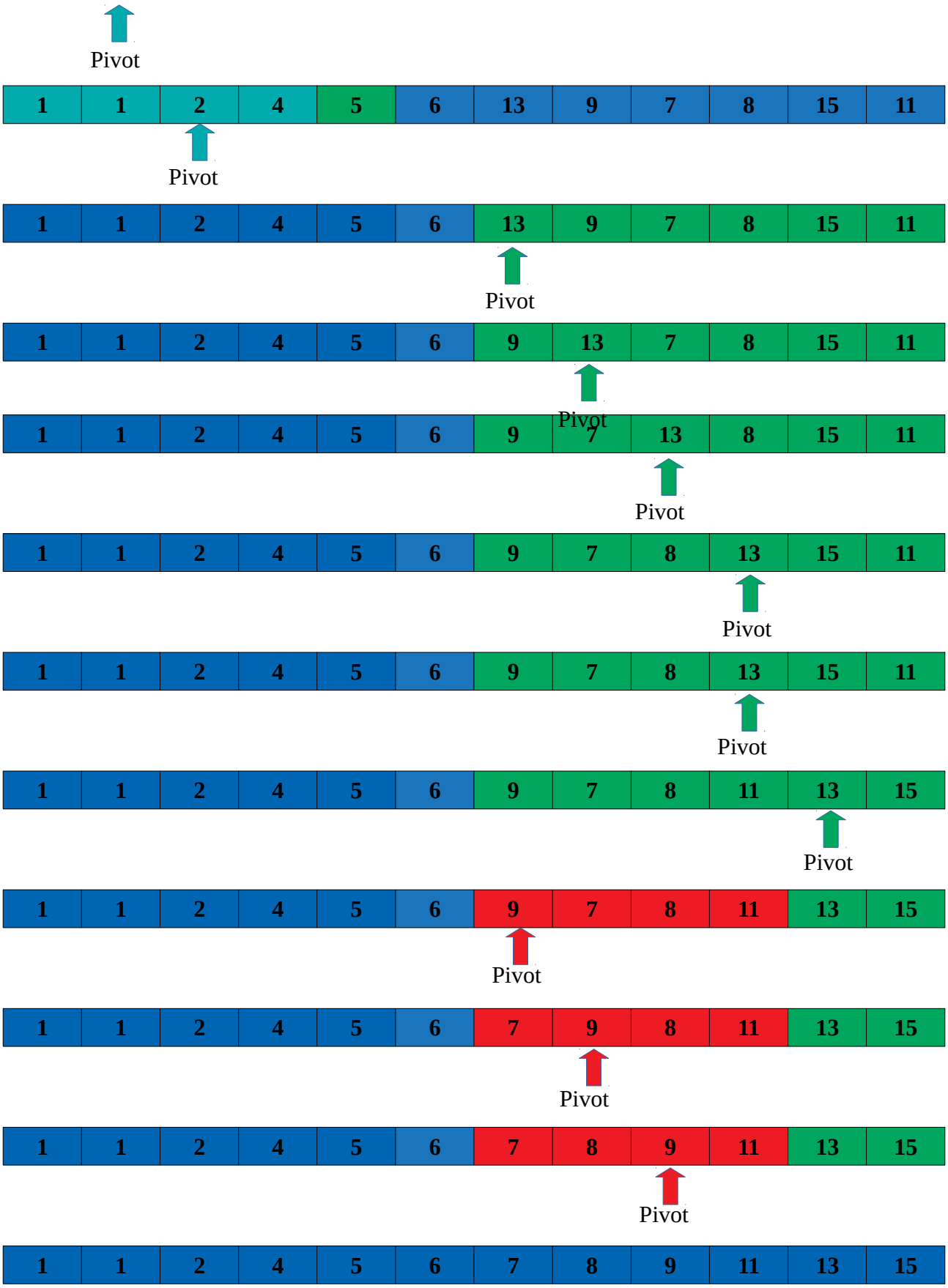




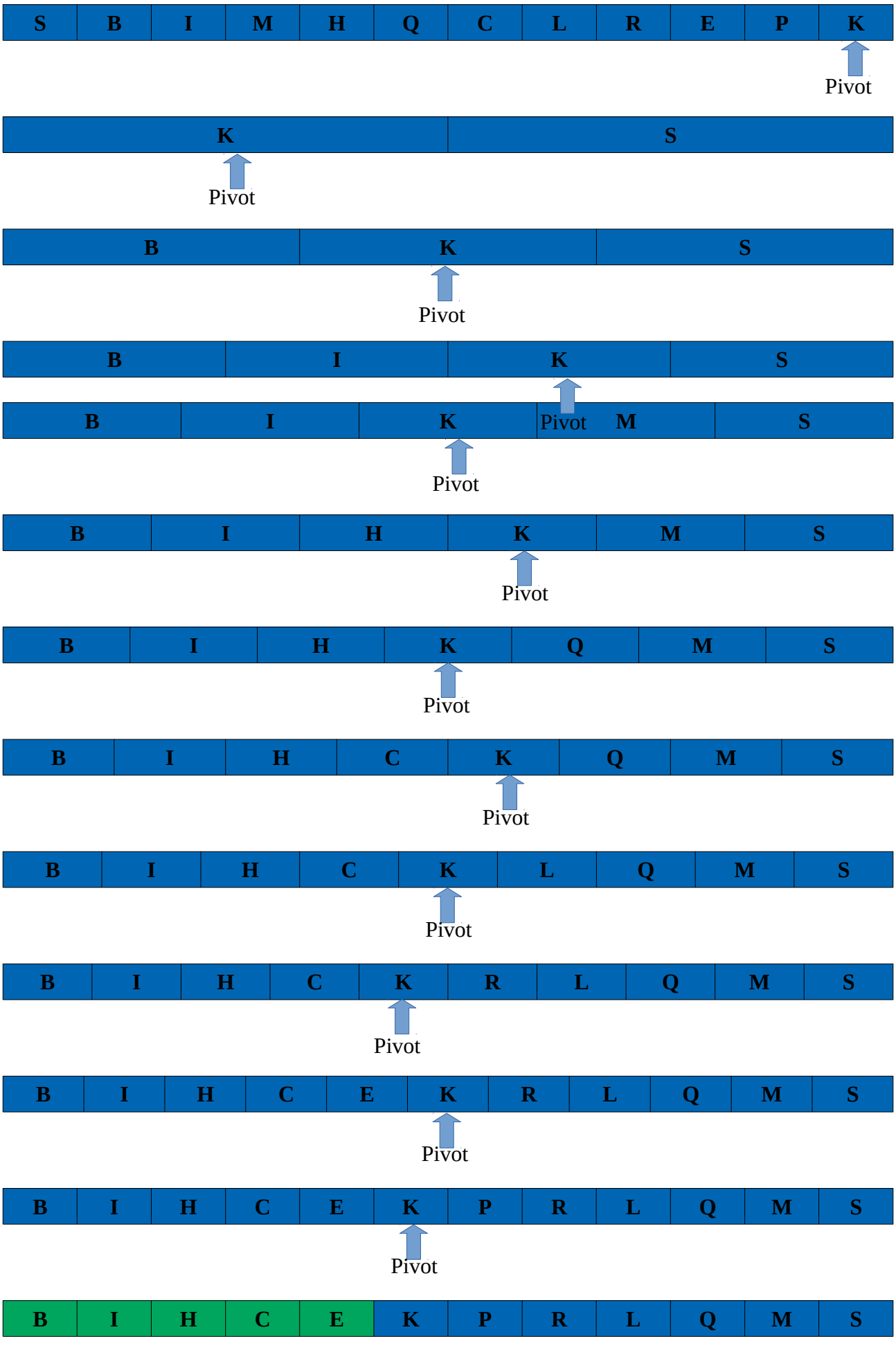


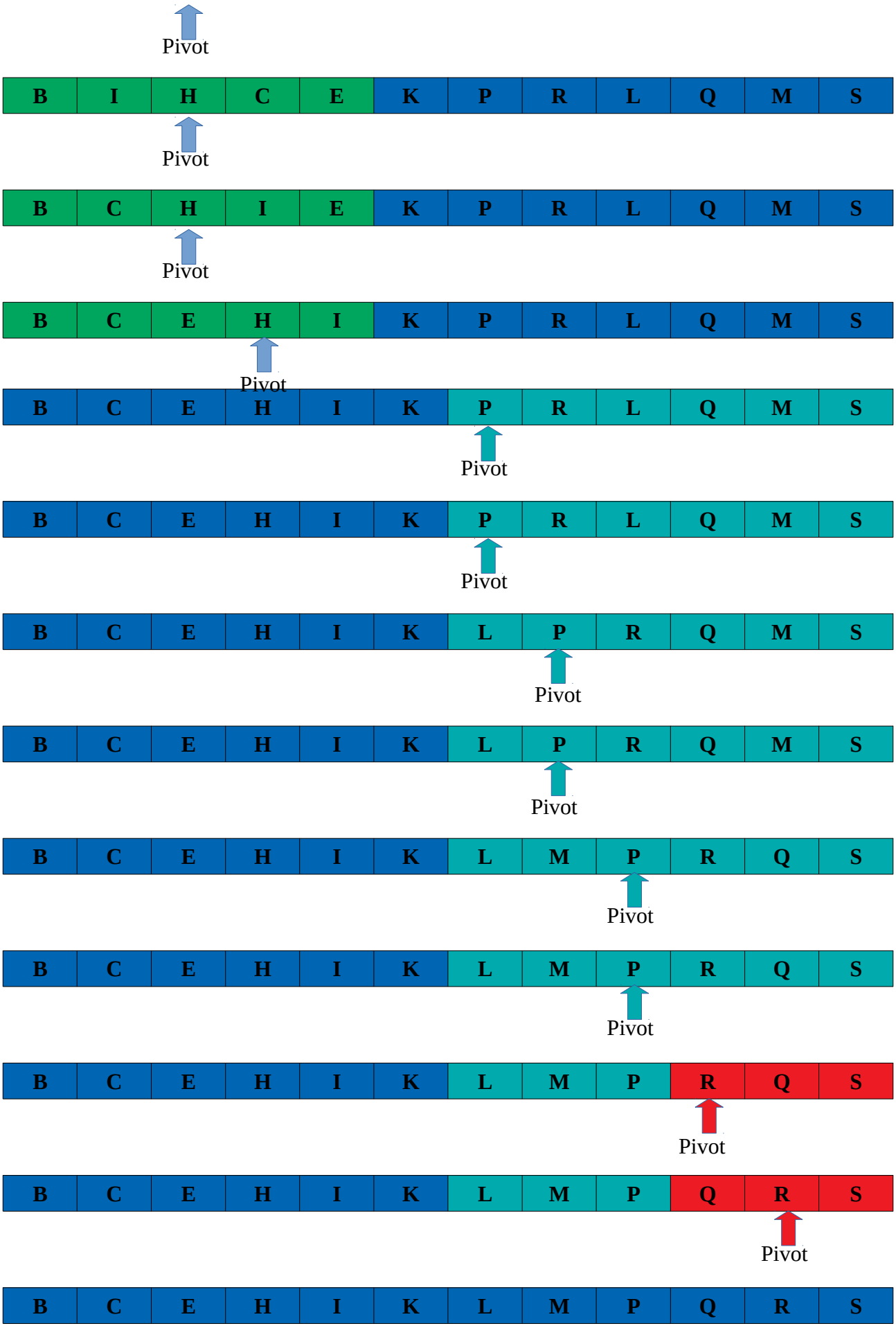



5 2 1 6 13 9 7 8
Pivot
5 2 1 1 6 13 9 7 8
Pivot
5 2 1 1 6 13 9 7 8 15
Pivot
5 2 1 1 4 6 13 9 7 8 15
Pivot
5 2 1 1 4 6 13 9 7 8 15 11
Pivot
5 2 1 1 4 6 13 9 7 8 15 11
Pivot
2 5 1 1 4 6 13 9 7 8 15 11
Pivot
2 1 5 1 4 6 13 9 7 8 15 11
Pivot
2 1 1 5 4 6 13 9 7 8 15 11
Pivot
2 1 1 4 5 6 13 9 7 8 15 11
Pivot
2 1 1 4 5 6 13 9 7 8 15 11
Pivot
1 2 1 4 5 6 13 9 7 8 15 11



*D={‘S’,’B’,’I’,’M’,’H’,’Q’,’C’,’L’,’R’,’E’,’P’,’K’}


S B I M H Q C L R E P K
Pivot
K S
Pivot
B K S
Pivot
B I K S
Pivot
B I K Pivot M S
Pivot
B I H K M S
Pivot
B I H K Q M S
Pivot
B I H C K Q M S
Pivot
B I H C K L Q M S
Pivot
B I H C K R L Q M S
Pivot
B I H C E K R L Q M S
Pivot
B I H C E K P R L Q M S
Pivot
B I H C E K P R L Q M S


Pivot
B I H C E K P R L Q M S
Pivot
B C H I E K P R L Q M S
Pivot
B C E H I K P R L Q M S
Pivot
B C E H I K P R L Q M S
Pivot
B C E H I K P R L Q M S
Pivot
B C E H I K L P R Q M S
Pivot
B C E H I K L P R Q M S
Pivot
B C E H I K L M P R Q S
Pivot
B C E H I K L M P R Q S
Pivot
B C E H I K L M P R Q S
Pivot
B C E H I K L M P Q R S
Pivot
B C E H I K L M P Q R S