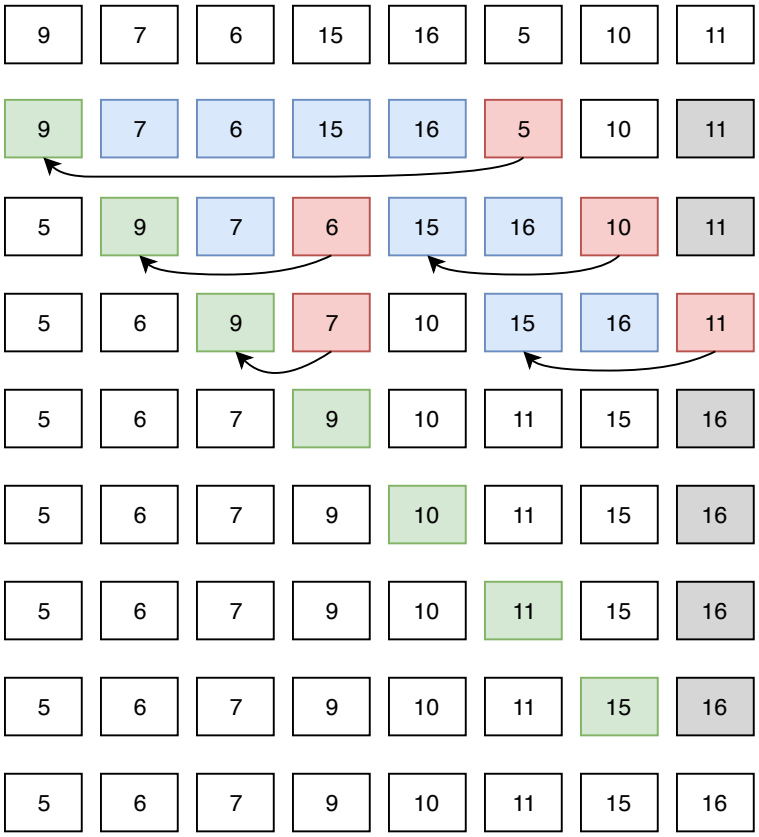
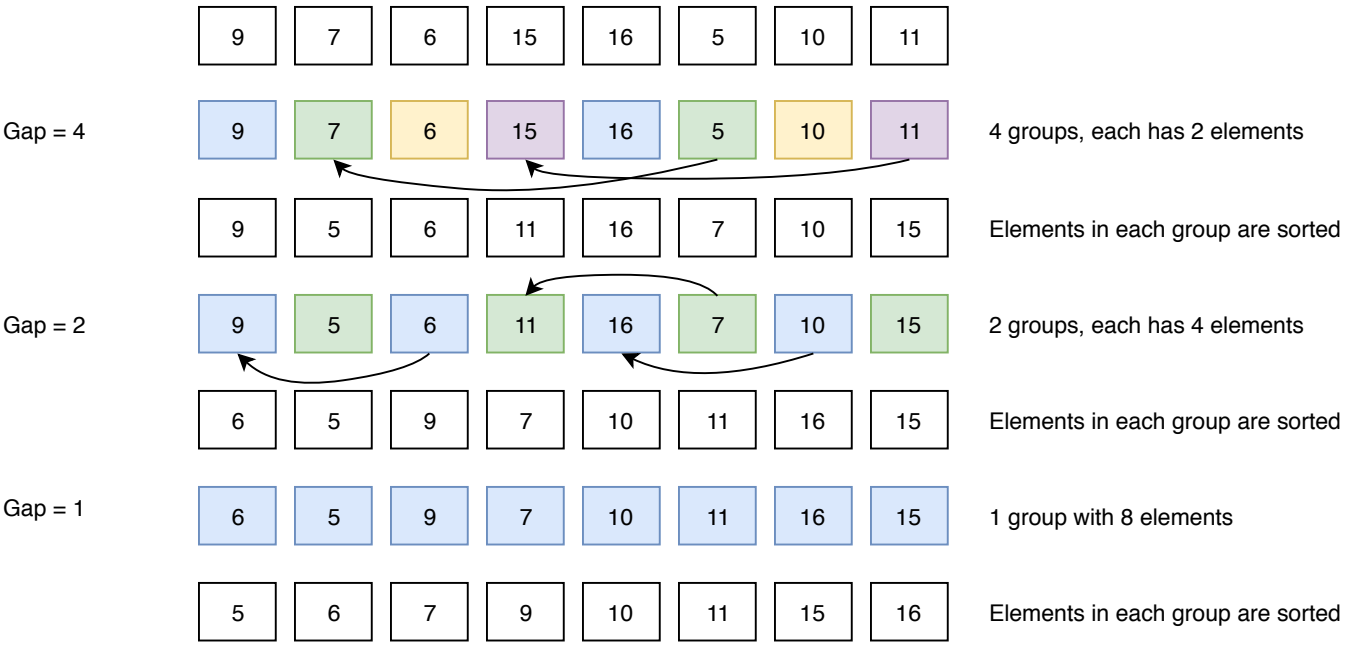
Bubble Sort

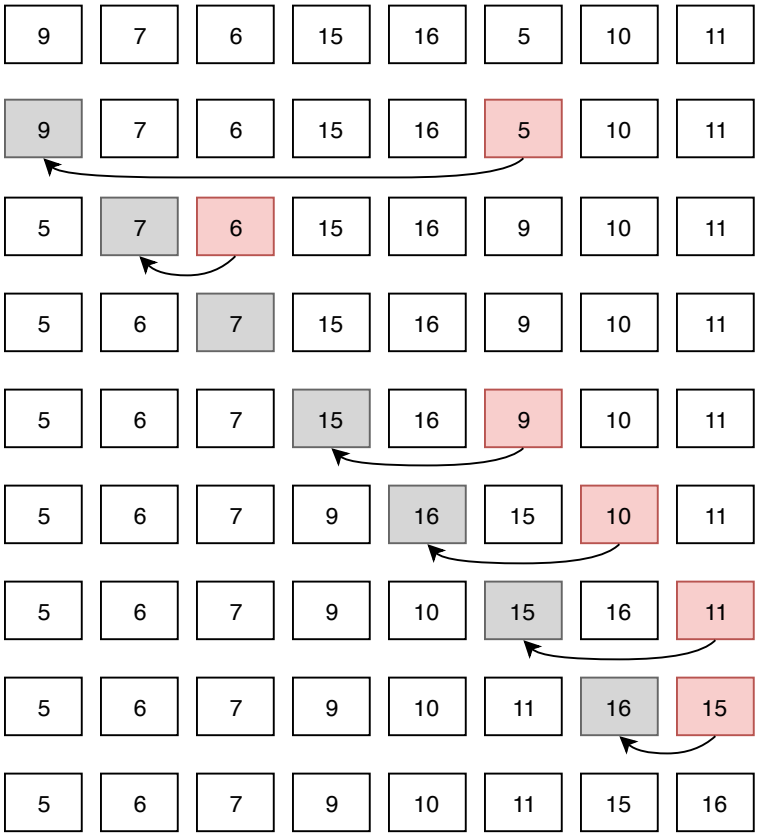


Insertion Sort

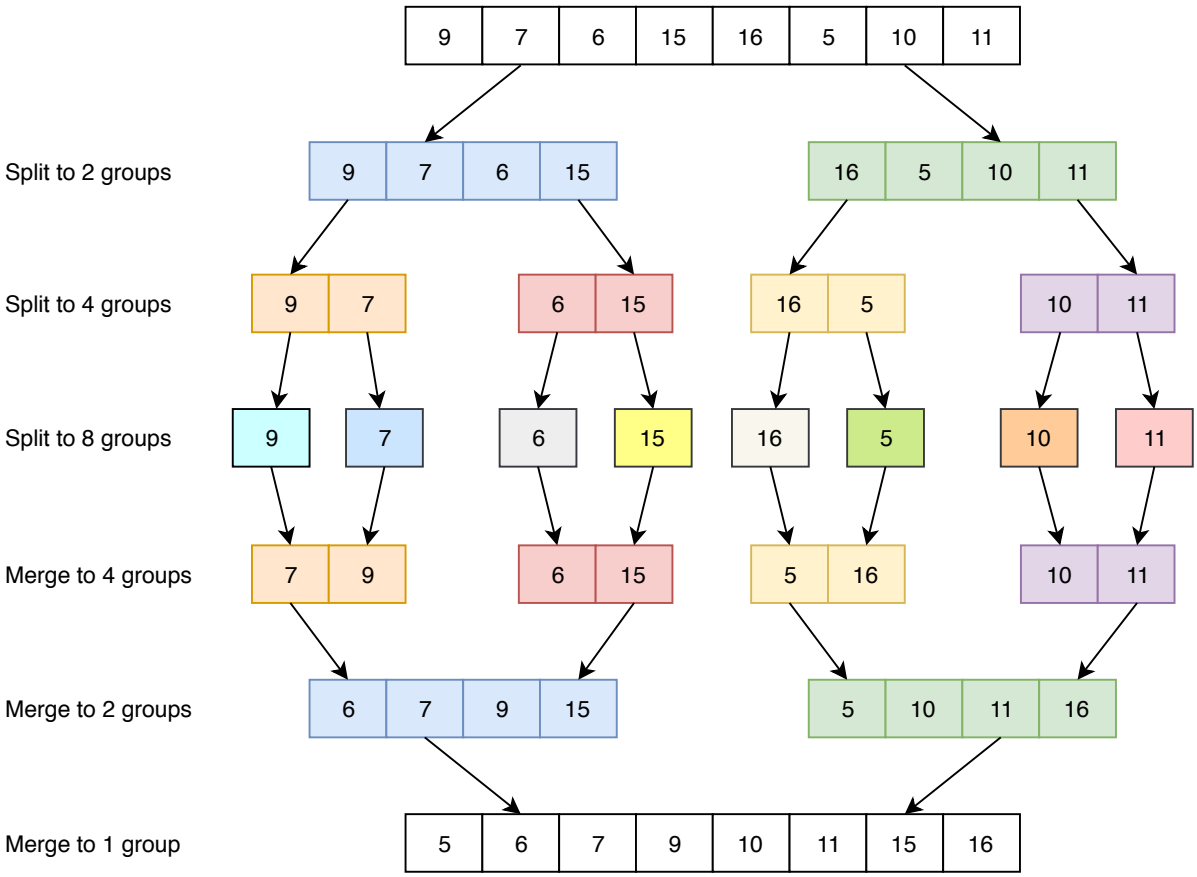
Shell Sort



Selection Sort



Merge Sort



Bucket Sort

9	7	6	15	16	5	10	11

a) find the maximum value of the given array

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

b) create and initialize the buckets, the size should be maximum + 1

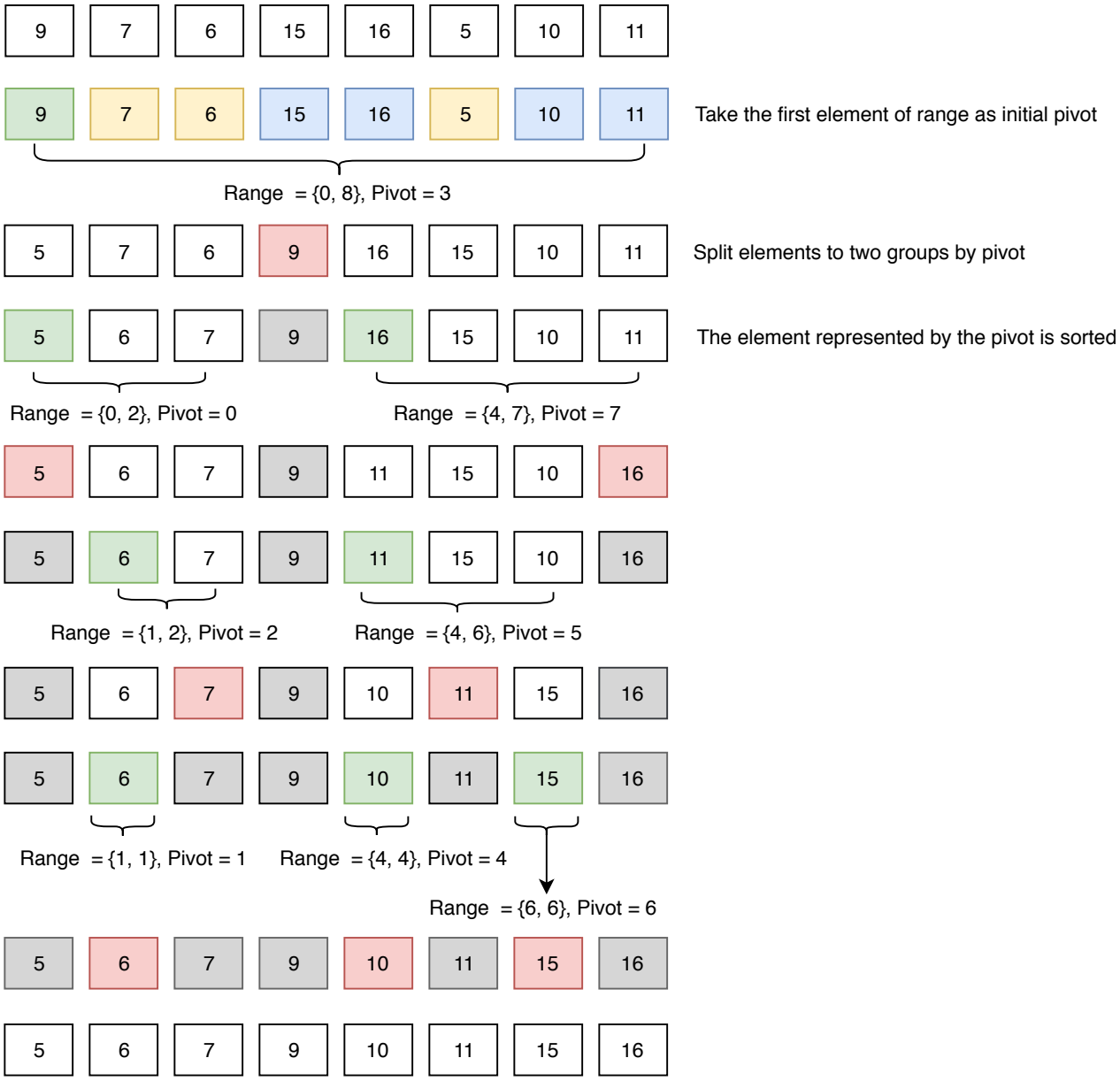
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	1	1

c) go through the original array, update corresponding bucket if that value exists

5	6	7	9	10	11	15	16

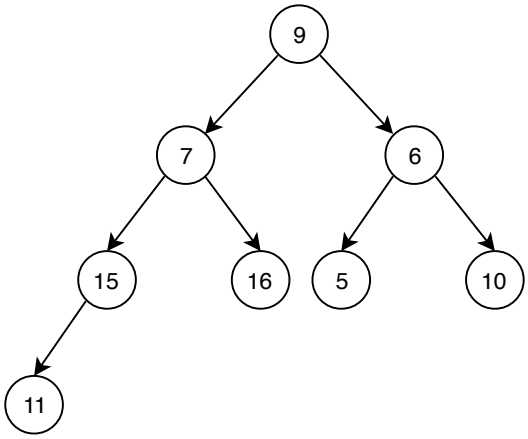
d) go through buckets from the smallest index, build the sorted array

Quick Sort


9 7 6 15 16 5 10 11
9 7 6 15 16 5 10 11
Take the first element of range as initial pivot
Range = {0, 8}, Pivot = 3
5 7 6 9 16 15 10 11
Split elements to two groups by pivot
5 6 7 9 16 15 10 11
The element represented by the pivot is sorted
Range = {0, 2}, Pivot = 0
Range = {4, 7}, Pivot = 7
5 6 7 9 11 15 10 16
5 6 7 9 11 15 10 16
Range = {1, 2}, Pivot = 2
Range = {4, 6}, Pivot = 5
5 6 7 9 10 11 15 16
5 6 7 9 10 11 15 16
Range = {1, 1}, Pivot = 1
Range = {4, 4}, Pivot = 4
Range = {6, 6}, Pivot = 6
5 6 7 9 10 11 15 16
5 6 7 9 10 11 15 16

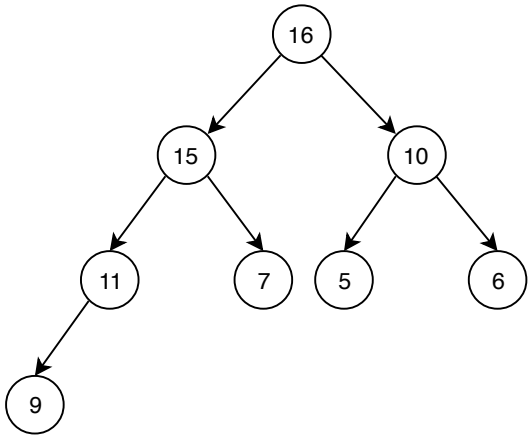
Heap Sort

Build Heap



9	7	6	15	16	5	10	11

Convert to Max Heap



16	15	10	11	7	5	6	9