arr = [3, 5, 2, 9, 12, 1, 8, 4]

function findMinMax(arr):

minVal = arr[0]

maxVal = arr[0]

for i from 1 to length of arr - 1:

if arr[i] < minVal:

minVal = arr[i]

else if arr[i] > maxVal:

maxVal = arr[i]

return minVal, maxVal

minVal, maxVal = findMinMax(arr)

Best case

# of comparisons – (n-1)

Worst case

# of comparisons – 2\* (n-1)

function minimax(arr, start, end):

if start == end:

// If only one element, return its index

return start

else if end - start == 1:

// If only two elements, compare and return indices of min and max accordingly

if arr[start] < arr[end]:

return start, end

else:

return end, start

else:

// If more than two elements, recursively divide and conquer

mid = (start + end) / 2

leftMin, leftMax = minimax(arr, start, mid)

rightMin, rightMax = minimax(arr, mid+1, end)

return min(leftMin, rightMin), max(leftMax, rightMax)

Recursive relation

T(n) = 2 \* T(n/2) + 2 ; where T(n) = 1 for n <= 2

Time complexity – O(n)