**Binary search:**

**Find the minimum depth of binary search tree**

Log2y = x

30/64 2

**Merge Sort:**

Assume that a mergesort algorithm in the worst case takes 30s for an input of size 64.

What is the max input size of a problem can be solved in 6 minutes?

O(n2)

O(642)

*12 . O(642) = 30 . 12*

*642 . x = 360*

*8 . 8 . 8 . 8*

*642 = 30*

*X2 = 360*

*642 / 30 = x2 / 360*

*642 = 30x2 / 360*

*642 = x2 / 12*

*642 = 642 . 12*

*X = sqrt 642 . 12*

*X = 64 sqrt 12*

*X = 217.6*

Worst-case performance O(n log n)

Best-case performance O(n log n) typical, O(n) natural variant

Average performance O(n log n)

Worst-case space complexity О(n) total with O(n) auxiliary, O(1) auxiliary with linked lists[1

*64 log2 64 = 30*

*x log2 x = 360*

*384 / 30 = x log2 x / 360*

*384 = x log2 x / 12*

*4608 = x log2 x*

*104608 = xx*

ANSWER: 512

**In an integer array, w/ N elements (N is large) find the minimum k elements ( k << N)**

**Solve using all merge sort, binary search, quick sort, heap sort, discover complexities**

**Quick Sort:**

**Given a list of #s: [1, 20, 11, 5, 2, 9, 16, 14, 13, 19] what would be the first pivot value using the medium of 3 methods: median of 3: choosing median of 3 values in array as pivot.**

**A.1**

**B.9**

**C.16**

**D.19**

**Write down complexities (both time/space) for merge sort, binary sort, bubble sort, heap sort, insertion sort, interpolation search**

**Given a sorted array of a # x, find the pair in array whose sum is closet to x.**