BINARY SEARCH

X = [1 5 7 9 11 15]

low = 0 mid = 3 high = 5

n = size of sublist = high − low + 1
= 6

mid = mid-point = low + n//2 = 3

binsearch (17, x)

while (low < high) :

STEPS :

1. low = 0, high = 5
n = high − low + 1 = 6
mid = low + n//2 = 3

splits the list X into two

X[3] = 17 ? No !

X[3] < 17
So move to the "right-hand" sublist

2. low = mid + 1 = 4
high = 5 (stays same)
n = high − low + 1 = 2
mid = low + n//2 = 5

[1 5 7 9 11 15]

mid = 5 high = 5.

low = 4

$$n = high - low + 1 = 2$$

$$mid = low + n // 2 = 5$$

x[5] = 17? No!

x[5] < 17.

So again, go to right sublist.

3. $low = mid + 1 = 5 + 1 = 6$

$high = 5$ (stays same)

$n = high - low + 1 = 0$

$mid = low + n // 2 = 6$

on next entry into loop condition

(low < high)

is violated. So exit loop!

binsearch (1, x)

STEPS :

1. low = 0, high = 5, n = 6

mid = low + n//2 = 3.

x[3] = 1 ? No !

[1   5   7   9   11   15]

low   mid   high

$\underline{\underline{n = 6}}$

x[3] > 1

So move to "left-hand" sublist

2. low = 0 (stays same)

high = mid − 1 = 2

n = high − low + 1 = 3

mid = low + n//2 = 0 + 3//2 = 1

[1   5   7   9   11   15]

low   mid   high

$\underline{\underline{n = 3}}$

$x[1] = 1$? No!

$x[1] > 1$, so move to left sublist

3. $low = 0$ (stays same)

$high = mid - 1 = 1 - 1 = 0$

$n = high - low + 1 = 1$

$mid = low + n // 2 = 0$

[1 5 7 9 11 15]

high

low

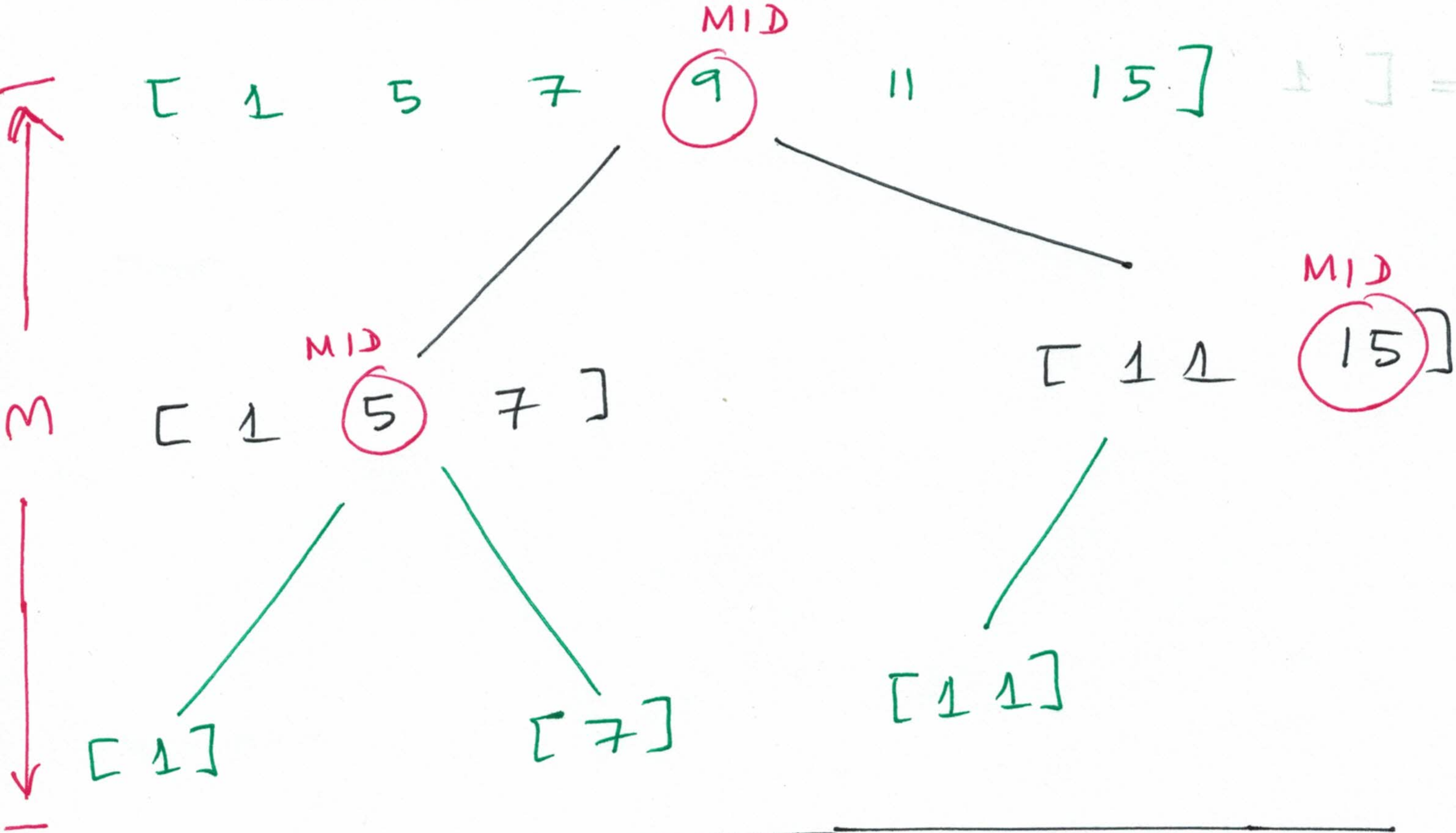
mid

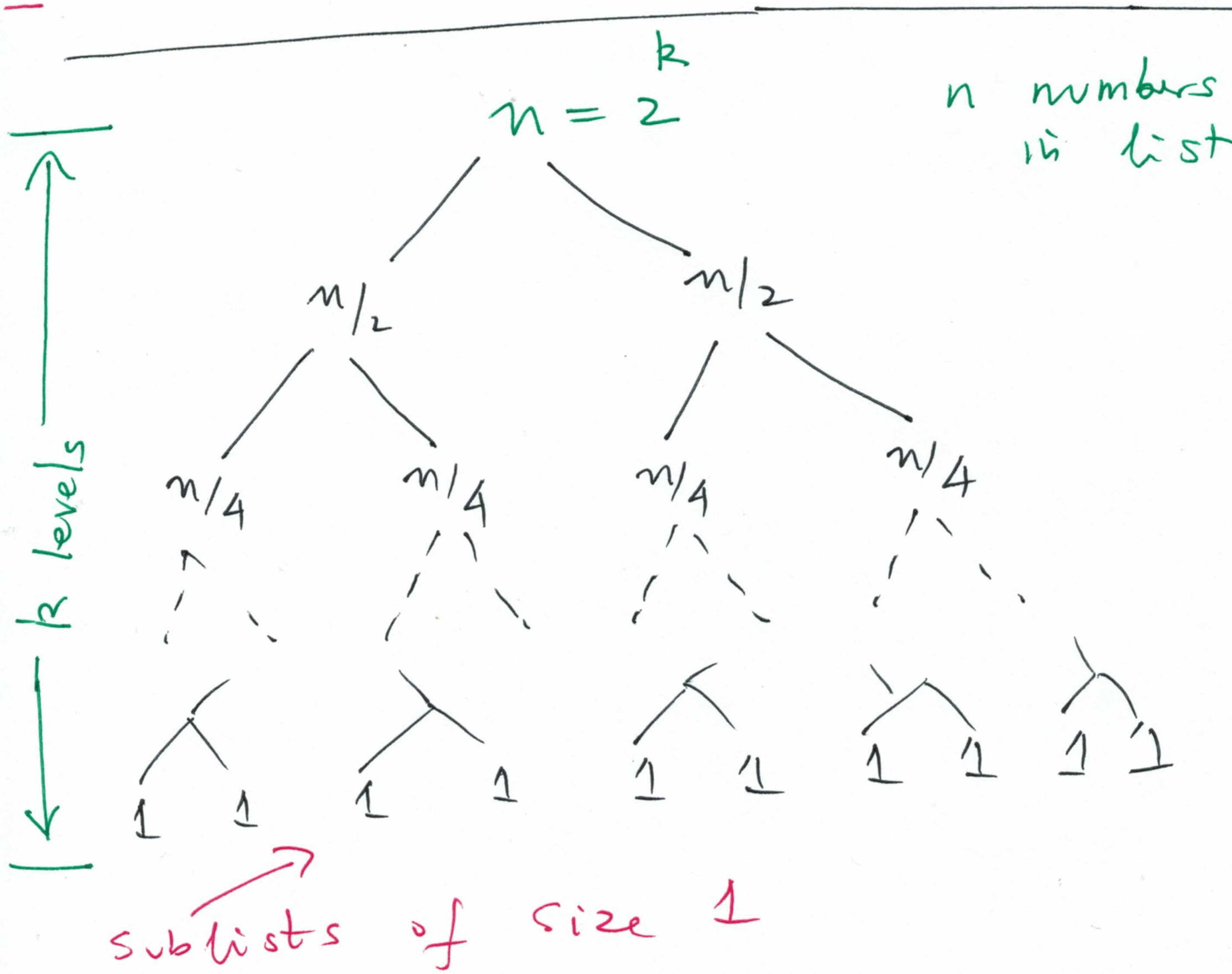
$\underline{\underline{n = 1}}$

$x[0] = 1$? YES!

So return 0 (index of 1) //

COMPLEXITY


MID
[1 5 7 9 11 15]
MID
[1 5 7]
MID
[11 15]
m
[1]
[7]
[11]


k
n = 2
n numbers in list
n/2
n/2
n/4
n/4
n/4
n/4
1 1 1 1 1 1 1 1 1 1
k levels
sublists of size 1

⇒ For list with n elements

# of required comparisons

is $\log_2 n$

ie. Complexity is $O(\log_2 n)$ //

[compare that to $O(n)$

comparisons reqd. in

Linear Search].