Algorithm Analysis and Data Structures
CS 5343.001: Homework #4

Due on Monday October 10, 2016 at 11:59pm

Professor Greg Ozbirn

Lizhong Zhang **(lxz160730)**
lxz160730@utdallas.edu

Contents

Problem 1

Part (a)

a c e d q n r w s

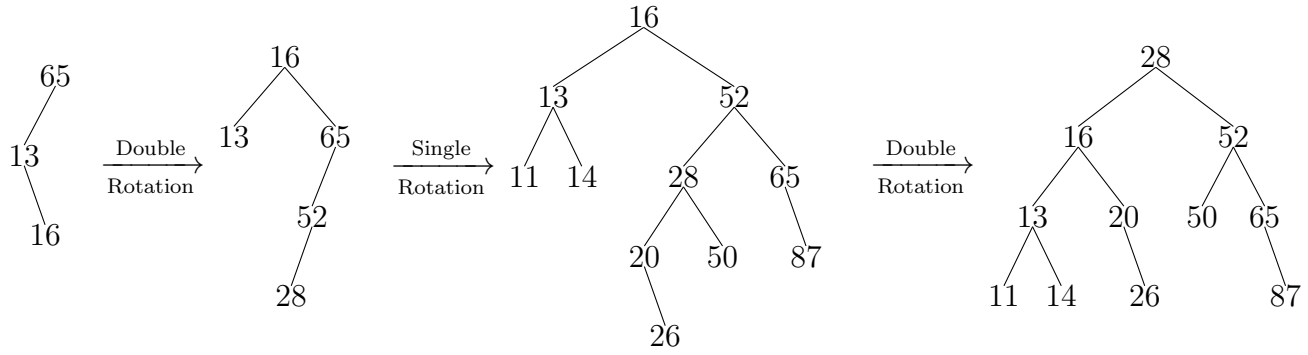
Part (b)

q e c a d r n s w

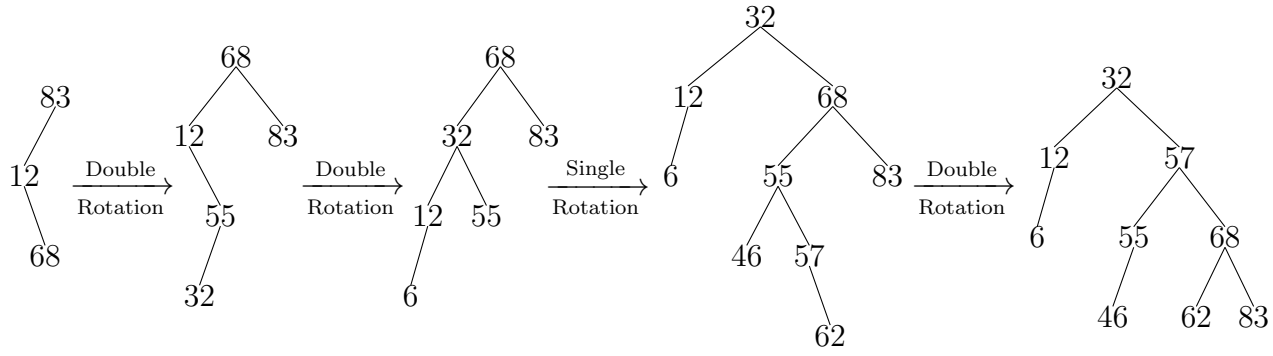
Part (c)

a c d e n w s r q

Problem 2

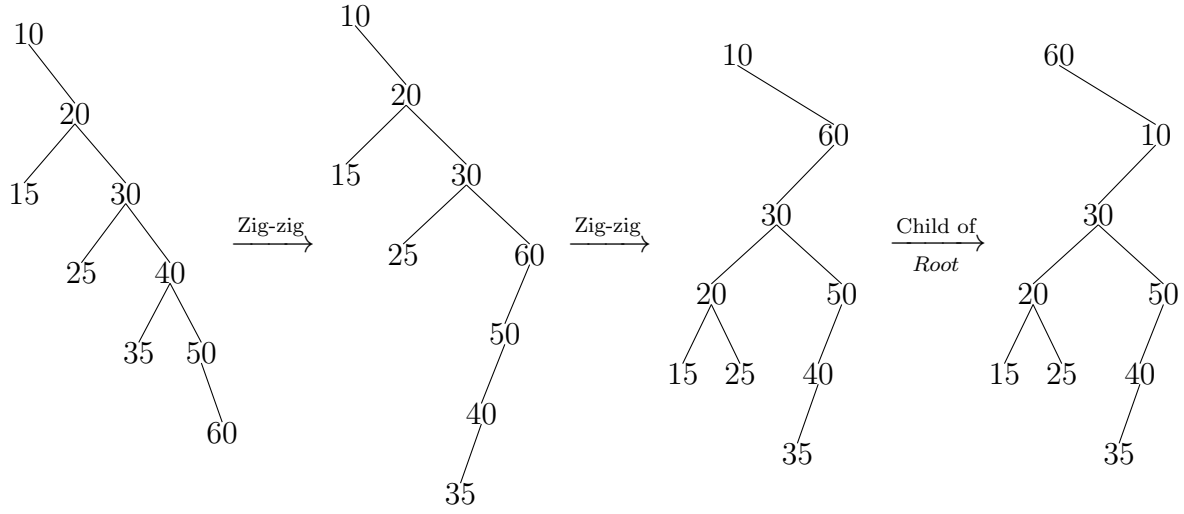


Problem 3



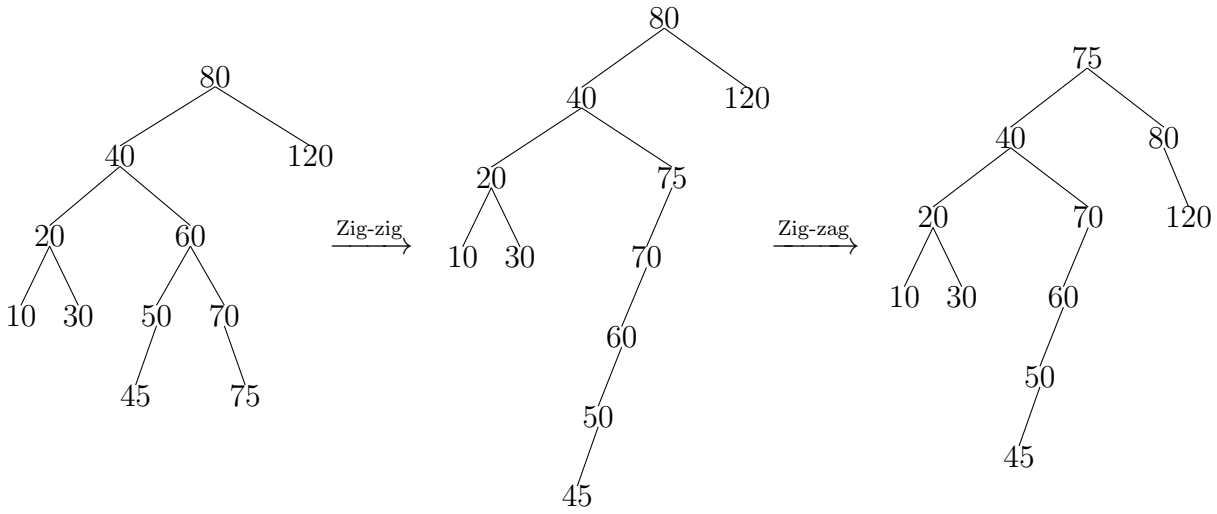
Problem 4

Access 60.

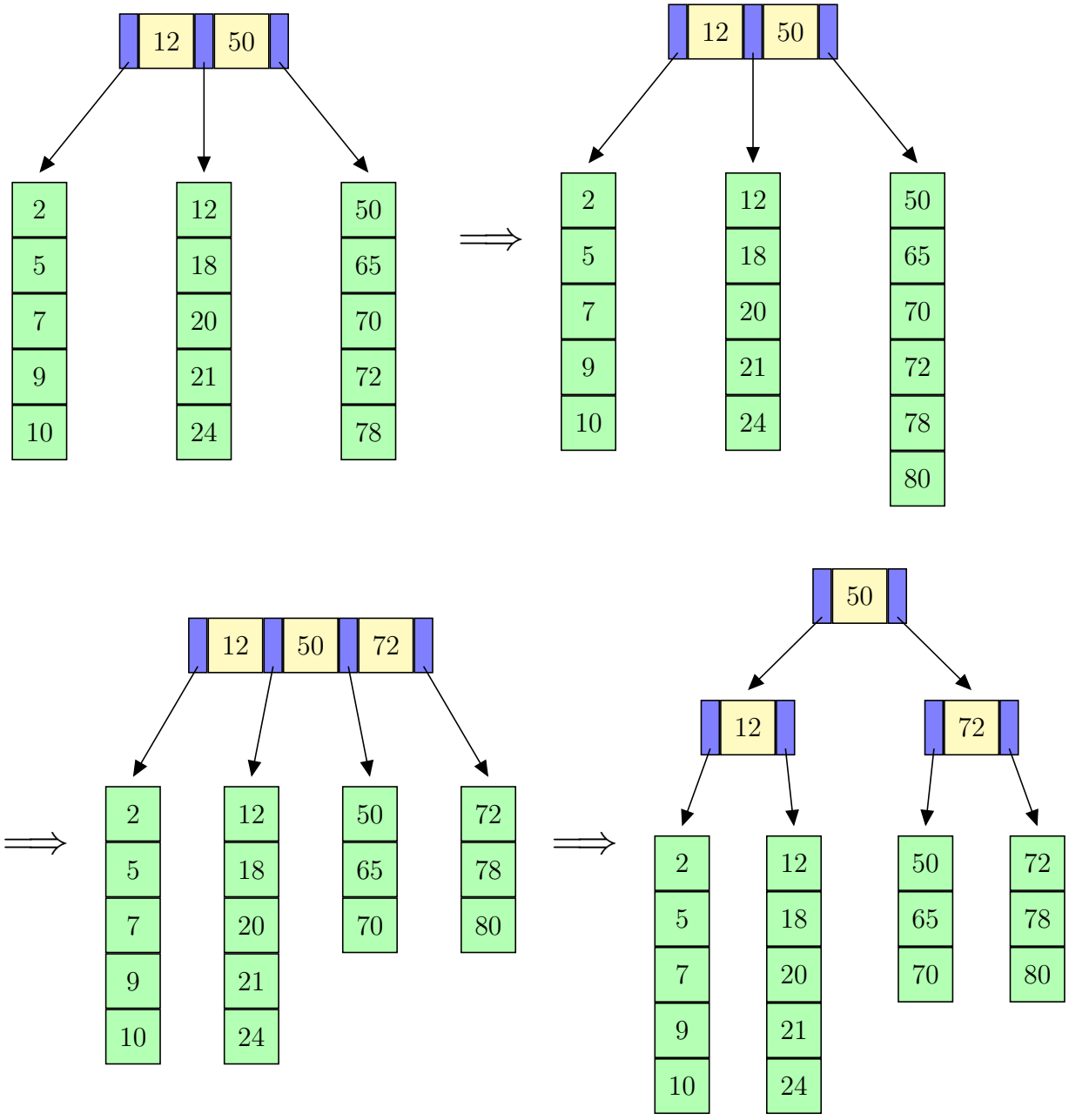


Problem 5

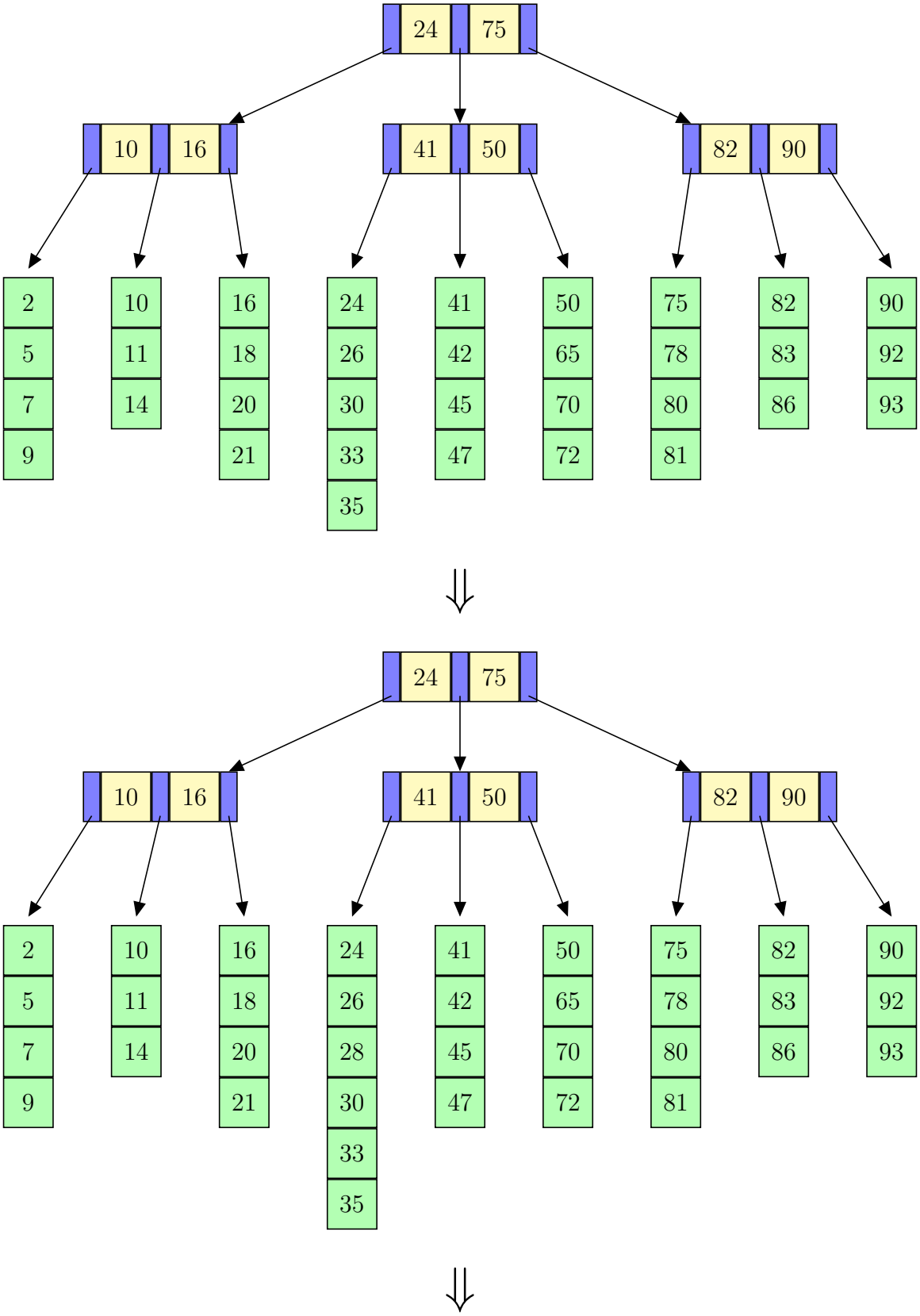
Access 75.



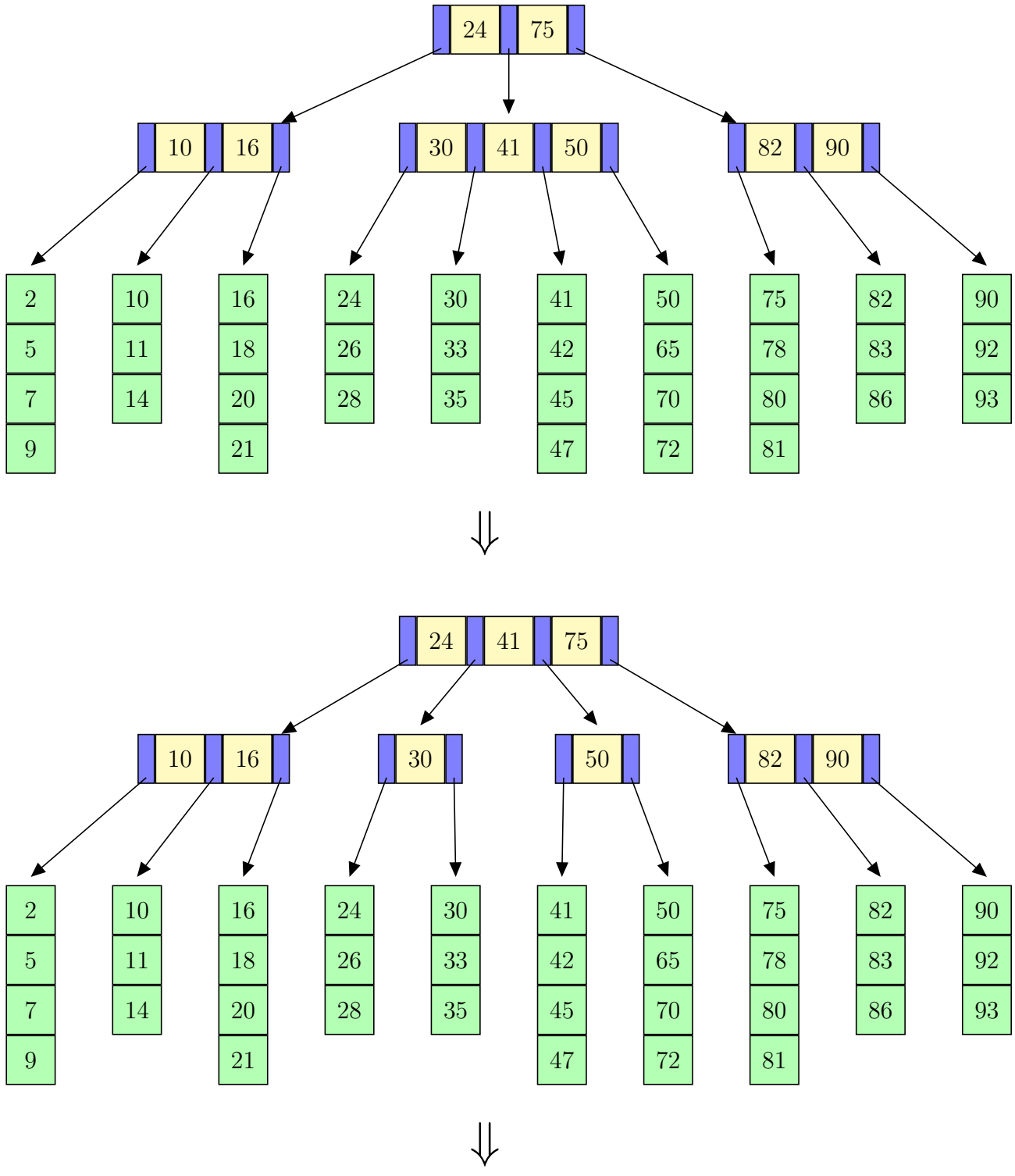
Problem 6

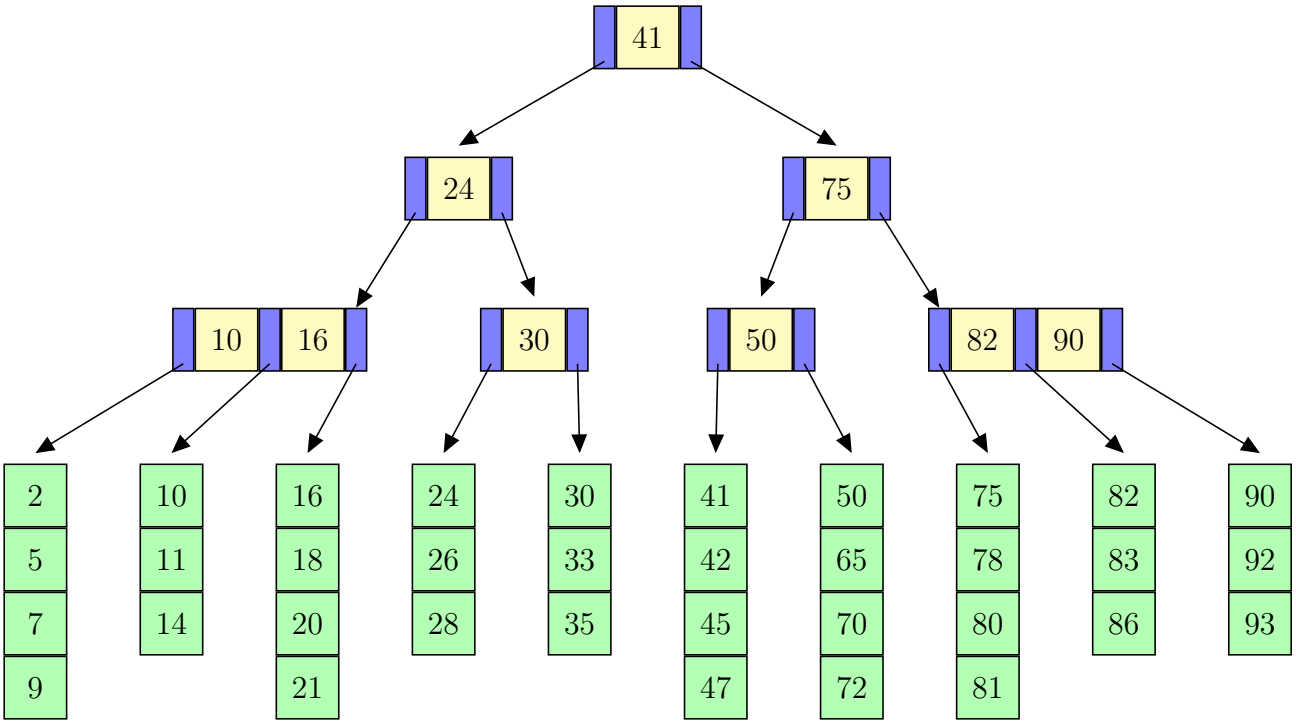


Problem 7



Problem 7 continued on next page...


24
75
10
16
30
41
50
82
90
2
5
7
9
10
11
14
16
18
20
21
24
26
28
30
33
35
41
42
45
47
50
65
70
72
75
78
80
81
82
83
86
90
92
93
24
41
75
10
16
30
50
82
90
2
5
7
9
10
11
14
16
18
20
21
24
26
28
30
33
35
41
42
45
47
50
65
70
72
75
78
80
81
82
83
86
90
92
93



Problem 8

If each key is 4 bytes, and there are $M-1$ keys, and each of the M pointers takes 4 bytes, then each node uses $4(M-1)+4M = 4M$ bytes. Solving for M gives $M = (3096+4)/8 = 38.75$, so $M = 38$. $L = 3096/36 = 86$

Problem 9

Since each leaf could be half full, $8,600,000$ records could take $8600000/(86/2) = 200000$ leaves.

Problem 10

In a binary tree, when there is only one node, it has two null pointers. Then add one node in the tree as a child node every time, it will produce two new null pointers. So if a binary tree has N nodes, it will have $2-(N-1)+2(N-1) = N+1$ null child pointers.

Problem 11

In a perfect binary tree which has N nodes(one filled at every level), when add another level

Problem 11 continued on next page...

in the tree, the number of nodes will be $2N + 1$.