Problem 1

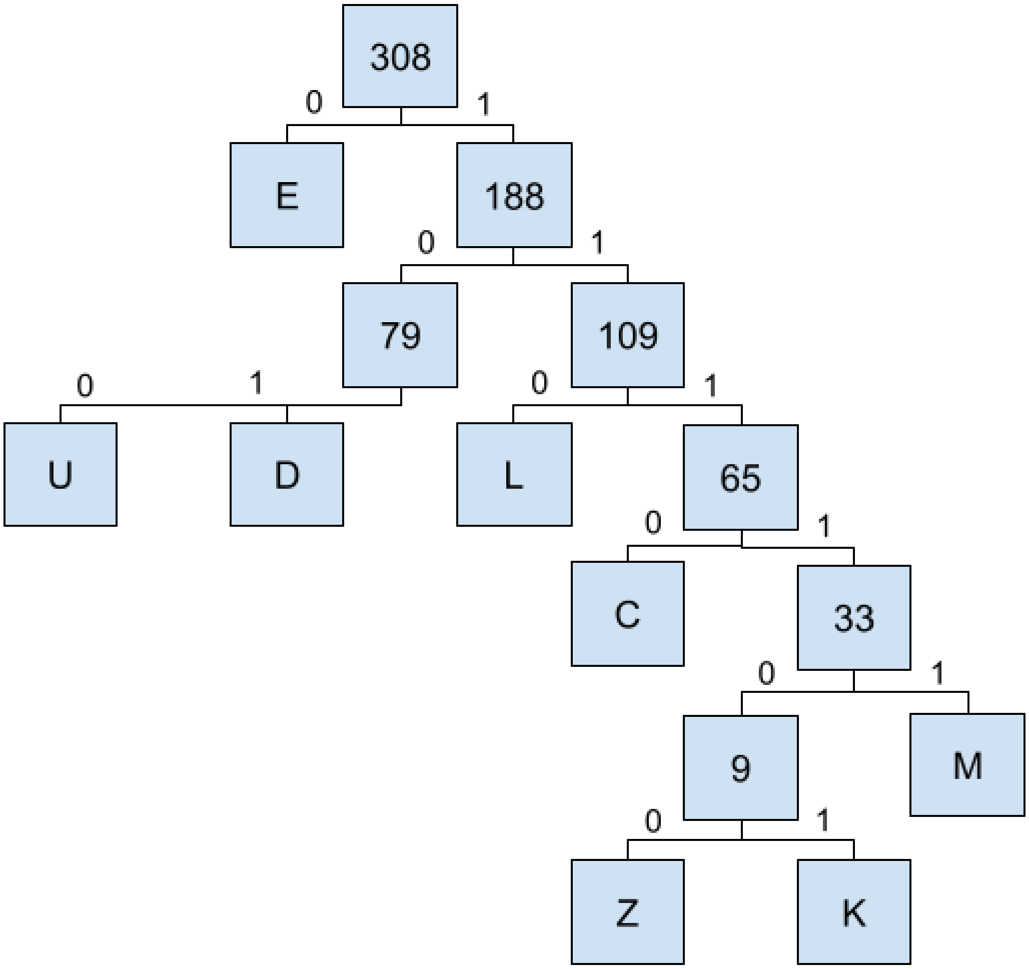
As no "node" is called more than once, the time complexity of the dynamically programmed Fibonacci is reduced to O(N). The recursive approach has a time complexity of O(2^N), which grows exponentially more to compute a sequence in comparison to the dynamic approach. In the provided table, the time complexity and measured time (in microseconds) by number of iterations, n, is given for each method. As expected, the recursive approach grows in time taken significantly more than the dynamic approach.

Method, f(n)	**Time Complexity**	**n=5**	**n=10**	**n=20**	**n=30**	**n=40**
Recursive	O(2^N)	55	18	268	23869	613707
Dynamic	O(N)	6	3	10	15	2

Problem 2

Rod Size	**Recursive Time**	**Recursive Max Revenue**	**Dynamic Time**	**Dynamix Max Revenue**
5	6 ms	10	5	10
10	16 ms	22	2	22
15	516 ms	35	3	35
20	0.0158 s	47	3	47
25	.332 s	60	5	60
30	10.171 s	72	6	72
35	N/A	N/A	8	85
40	N/A	N/A	9	97
45	N/A	N/A	11	110
50	N/A	N/A	15	122

Problem 3


308
0
1
E
188
0
1
79
109
0
1
0
1
U
D
L
65
0
1
C
33
0
1
9
M
0
1
Z
K