დათა ჭანუყვაძე | ჯგ.8

class Node:

def \_\_init\_\_(self, key):

self.l = None

self.r = None

self.v = key

def bin\_tree():

root = Node(1)

root.l = Node(2)

root.r = Node(3)

root.l.l = Node(4)

root.l.r = Node(5)

root.r.l = Node(6)

root.r.r = Node(7)

return root

def add\_elem(root, key):

new\_node = Node(key)

root.l.l = new\_node

def delete\_elem(root, key):

root.l.l = None

tree = bin\_tree()

add\_elem(tree, 10)

delete\_elem(tree, 5)

min\_elems = 3 # H+1

max\_elems = 7 # 2^H+1

min\_leaves = 4 # 2^H

max\_leaves = 2 # 2^H-1

def maximize\_profit(prices\_list, lengths\_list, target\_length):

max\_profit\_dp = [0] \* (target\_length + 1)

for current\_length in range(1, target\_length + 1):

for price\_index in range(len(prices\_list)):

if lengths\_list[price\_index] <= current\_length:

max\_profit\_dp[current\_length] = max(

max\_profit\_dp[current\_length],

max\_profit\_dp[current\_length - lengths\_list[price\_index]] + prices\_list[price\_index]

)

return max\_profit\_dp[target\_length]

prices = [1, 2, 3, 4, 5, 6, 7, 8]

lengths = [3, 5, 8, 9, 10, 17, 17, 20]

target\_length = 10

max\_profit = maximize\_profit(prices, lengths, target\_length)

print(f"len - {target\_length} , profit: {max\_profit}")