

108)Knapsack problem

CODE:

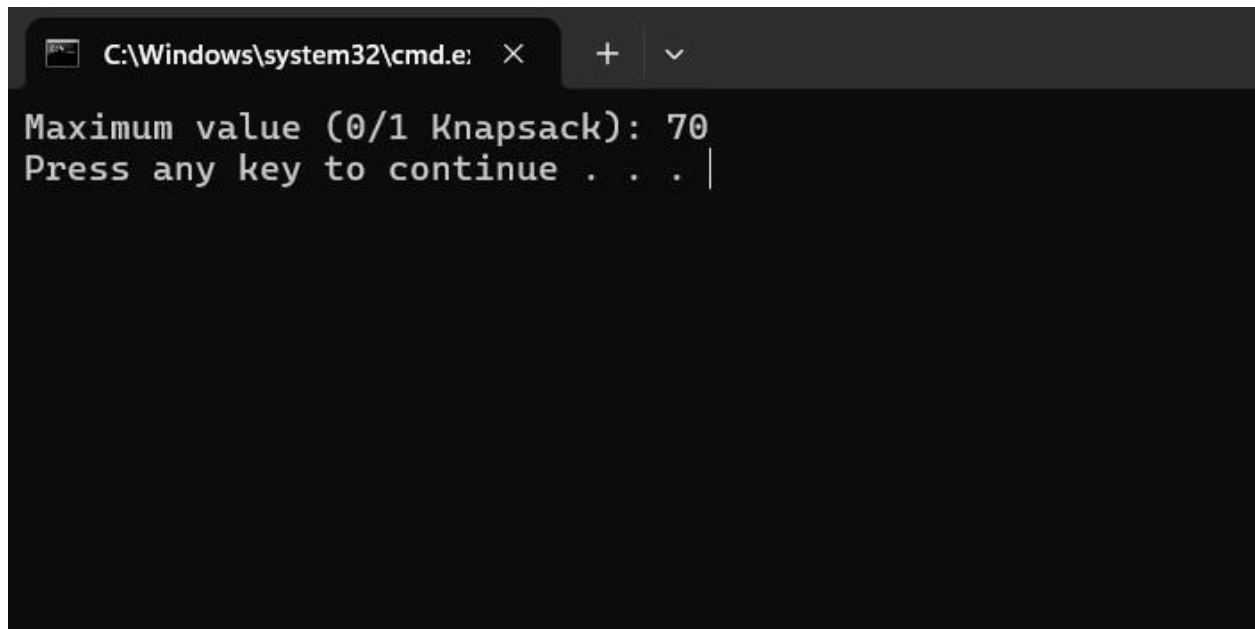
```
def knapsack_01(weights, values, capacity):
    n = len(weights)
    dp = [[0] * (capacity + 1) for _ in range(n + 1)]
    for i in range(1, n + 1):
        for w in range(capacity + 1):
            if weights[i] - 1] <= w:
                dp[i][w] = max(dp[i - 1][w], values[i - 1] + dp[i - 1][w - weights[i] - 1])
            else:
                dp[i][w] = dp[i - 1][w]

    return dp[n][capacity]
```

```
weights = [1, 2, 3, 4, 5] values
= [10, 20, 30, 40, 50] capacity
= 7
```

```
print("Maximum value (0/1 Knapsack):", knapsack_01(weights, values, capacity))
```

OUTPUT:

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.e' with a close button. The window content displays the output of the program: 'Maximum value (0/1 Knapsack): 70' followed by 'Press any key to continue . . . |' with a cursor at the end of the line.

TIME COMPLEXITY :  $O(n \log n)$