108)Knapsack problem

CODE:

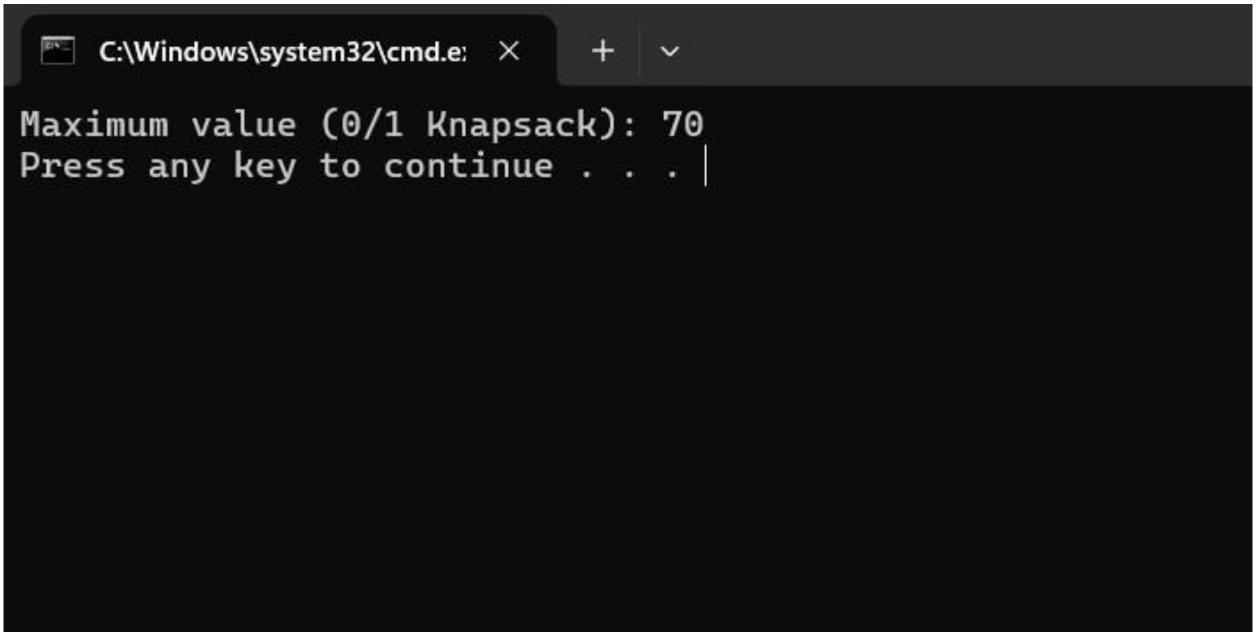
```
def knapsack_01(weights, values, capacity):
n = len(weights)
    dp = [[0] * (capacity + 1) for _ in range(n + 1)]
         for i in range(1, n + 1):         for w in range(capacity + 1):             if weights[i
- 1] <= w:                dp[i][w] = max(dp[i - 1][w], values[i - 1] + dp[i - 1][w -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



```
Maximum value (0/1 Knapsack): 70
Press any key to continue . . .
```

TIME COMPLEXITY : O(nlogn)