

6. Container Loading ,

Code:

```
def container_loading(weights, values, capacity):  
    ratio = [(values[i] / weights[i], weights[i], values[i]) for i in range(len(weights))]  
    ratio.sort(key=lambda x: x[0], reverse=True)  
    total_value = 0  
    total_weight = 0  
    for r in ratio:  
        if total_weight + r[1] <= capacity:  
            total_weight += r[1]  
            total_value += r[2]  
        else:  
            break  
    return total_value  
weights = [10, 20, 30]  
values = [60, 100, 120]  
capacity = 50  
print(container_loading(weights, values, capacity))
```

output:

```
PS C:\Users\karth>  
PS C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Documents/OriginLab/daa.py  
160  
PS C:\Users\karth> █
```

Time complexity:

$F(n) = O(nw)$