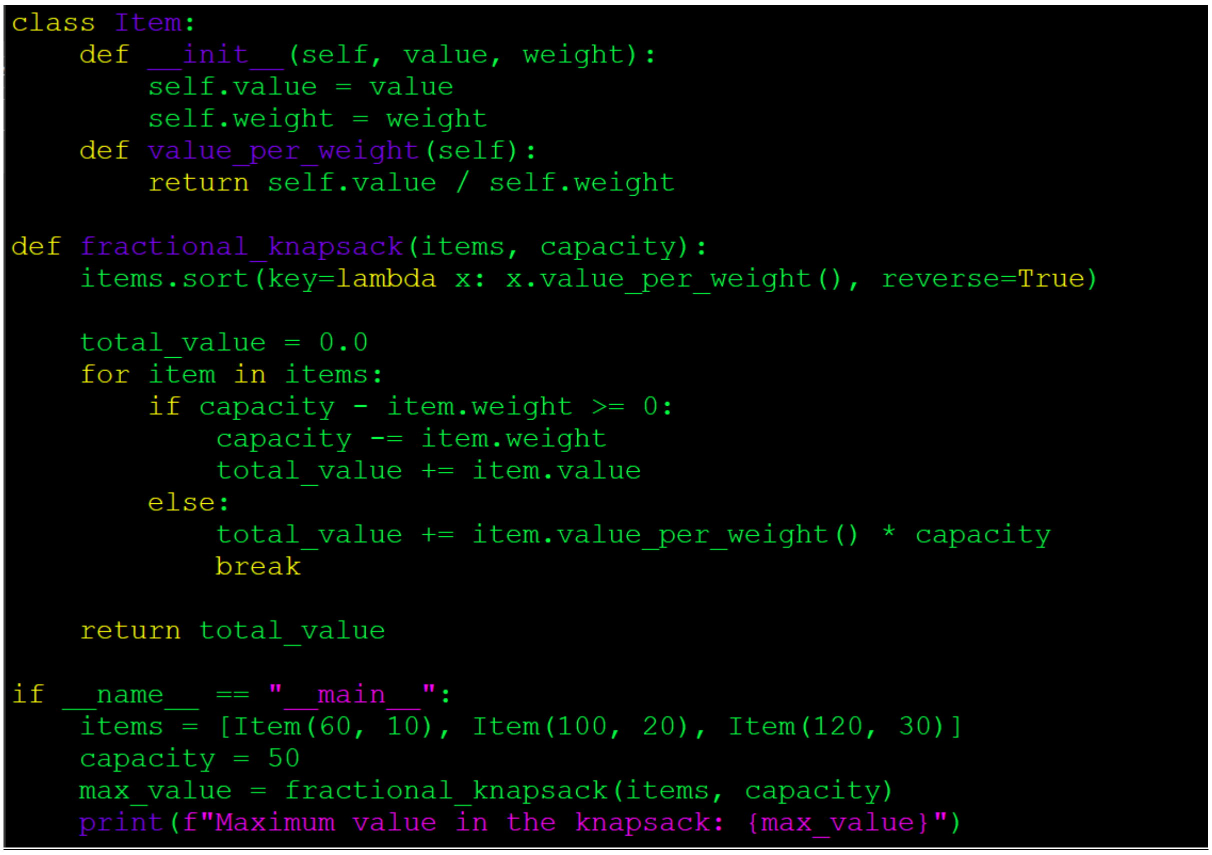
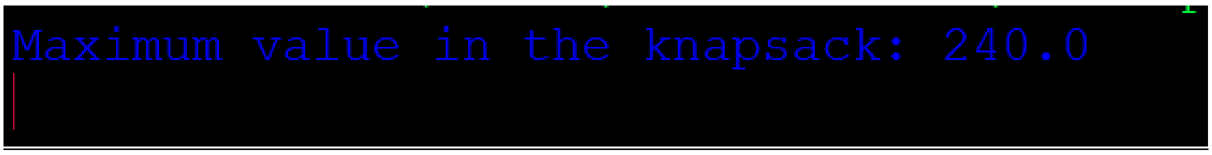
112. Knapsack problem using greedy

Code:



```python
class Item:
    def __init__(self, value, weight):
        self.value = value
        self.weight = weight
    def value_per_weight(self):
        return self.value / self.weight

def fractional_knapsack(items, capacity):
    items.sort(key=lambda x: x.value_per_weight(), reverse=True)

    total_value = 0.0
    for item in items:
        if capacity - item.weight >= 0:
            capacity -= item.weight
            total_value += item.value
        else:
            total_value += item.value_per_weight() * capacity
            break

    return total_value

if __name__ == "__main__":
    items = [Item(60, 10), Item(100, 20), Item(120, 30)]
    capacity = 50
    max_value = fractional_knapsack(items, capacity)
    print(f"Maximum value in the knapsack: {max_value}")
```

Output:



```
Maximum value in the knapsack: 240.0
```

Time Complexity:

- T(n)= O(n log n)