

object ▾

Scratches

armstrong non recursive.py

armstrong recursive.py

Binary search.py

copy the program non recursive.py

copy the program recursive.py

fact non recursive.py

fact recursive.py

fib non recursive.py

fib recursive.py

gcd non recursive.py

gcd recursive.py

knapsack.py

lcm non recursive.py

lcm recursive.py

max and min.py

max non recursive.py

max recursive.py

mergesort.py

MST.py

multiplication non recursive.py

multiplication recursive.py

palindrome non recursive.py

palindrome recursive.py

prime or not non recursive.py

cursive.py ×

Binary search.py ×

max and min.py ×

stressens multiplication.py ×

mergesort.py ×

MST.py ×

knapsack.py ×

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

```
class KnapsackPackage(object):  
    def __init__(self, weight, value):  
        self.weight = weight  
        self.value = value  
        self.cost = value / weight  
  
    def __lt__(self, other):  
        return self.cost < other.cost  
  
class FractionalKnapsack(object):  
    def __init__(self):  
        pass  
  
    def knapsackGreProc(self, W, V, M, n):  
        packs = []  
        for i in range(n):  
            packs.append(KnapsackPackage(W[i], V[i]))  
        packs.sort(reverse=True)  
        remain = M  
        result = 0  
        i = 0
```

KnapsackPackage

→

__lt__

Project ▾

Scratches

- armstrong non recursive.py
- armstrong recursive.py
- Binary search.py
- copy the program non recursive.py
- copy the program recursive.py
- fact non recursive.py
- fact recursive.py
- fib non recursive.py
- fib recursive.py
- gcd non recursive.py
- gcd recursive.py
- knapsack.py**
- lcm non recursive.py
- lcm recursive.py
- max and min.py
- max non recursive.py
- max recursive.py
- mergesort.py
- MST.py
- multiplication non recursive.py
- multiplication recursive.py
- palindrome non recursive.py
- palindrome recursive.py
- prime or not non recursive.py

knapsack.py

```
21     result = 0
22     i = 0
23     stopProc = False
24     while (stopProc != True):
25         if (packs[i].weight <= remain):
26             remain -= packs[i].weight
27             result += packs[i].value
28             print("Pack ", i, " - Weight ", packs[i].weight, " - Value ", packs[i].value)
29             if (packs[i].weight > remain):
30                 i += 1
31             if (i == n):
32                 stopProc = True
33     print("Max Value:\t", result)
34
35
36 ▶ if __name__ == "__main__":
37     W = [15, 10, 2, 4]
38     V = [30, 25, 2, 6]
39     M = 37
40     n = 4
41     proc = FractionalKnapsack()
42     proc.knapsackGreProc(W, V, M, n)
```

Project ▾

Scratches

- armstrong non recursive.py
- armstrong recursive.py
- Binary search.py
- copy the program non recursive.py
- copy the program recursive.py
- fact non recursive.py
- fact recursive.py

```
31 W = [15, 10, 2, 4]
32 V = [30, 25, 2, 6]
33 M = 37
34 n = 4
35 proc = FractionalKnapsack()
36 proc.knapsackGreProc(W, V, M, n)

if __name__ == "__main__":
```

scratch ×

C:\Users\kativ\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:/Users/kativ/AppData/Roaming/JetBrains/PyCharmCE2022.1/scratches/scratch.py

```
Pack 0 - Weight 10 - Value 25
Pack 1 - Weight 15 - Value 30
Pack 2 - Weight 4 - Value 6
Pack 3 - Weight 2 - Value 2
Max Value: 83

Process finished with exit code 0
```