

**Name: K.Charan Kumar Reddy**

**Roll No:21071A6724**

# **FLAT-Assignment**

## **Fractional Knapsack Problem**

Given the weights and values of  $N$  items, in the form of {value, weight} by putting these items in a knapsack of capacity  $W$ , we get the maximum total value in the knapsack. In **Fractional Knapsack**, we break items for maximizing the total value of the knapsack.

### **Algorithm-**

Fractional Knapsack (Array  $W$ , Array  $V$ , int  $M$ )

```
1. for i <- 1 to size (V)
2.     calculate cost[i] <-  $V[i] / W[i]$ 
3. Sort-Descending (cost)
4. i <- 1
5. while (i <= size(V))
6.     if  $W[i] \leq M$ 
7.          $M \leftarrow M - W[i]$ 
8.         total <- total +  $V[i]$ ;
9.     if  $W[i] > M$ 
10.        i <- i+1
```

### **Code-**

```
class Item:
    def __init__(self, value, weight):
        self.value = value
        self.weight = weight

def fractionalKnapsack(W, arr):

    arr.sort(key=lambda x: (x.value/x.weight), reverse=True)
    finalvalue = 0.0

    for item in arr:

        if item.weight <= W:
            W -= item.weight
            finalvalue += item.value

        else:
            finalvalue += item.value * W / item.weight
            break
```

```
    return finalvalue

if __name__ == "__main__":

    W = 50
    arr = [Item(60, 10), Item(100, 20), Item(120, 30)]
    max_val = fractionalKnapsack(W, arr)
    print(max_val)
```

### **Output-**

Maximum value we can obtain = 240