**Experiment No: 5**

Student Name and Roll Number: Jaspreet Singh 18CSU094

Semester /Section: CS-V-CSA

Date: 8th September 2020

Faculty Signature:

Remarks:

**Objective**:

To Implement Fractional Knapsack Problem using any Programming Language.

**Theory:**

* Given weights and values of n items, we need to put these items in a knapsack of capacity W to get the maximum total value in the knapsack.
* In Fractional Knapsack, we can break items for maximizing the total value of knapsack. This problem in which we can break an item is also called the fractional knapsack problem.
* An efficient solution is to use Greedy approach. The basic idea of the greedy approach is to calculate the ratio value/weight for each item and sort the item on basis of this ratio. Then take the item with the highest ratio and add them until we can’t add the next item as a whole and at the end add the next item as much as we can. Which will always be the optimal solution to this problem.

**Code:**

def knapsack(n, capacity, value\_list, weight\_list):

unitValues\_list = []

#First lets calculate the unitValues\_list

for i in range (n):

unitValue = (value\_list[i]/weight\_list[i])

unitValues\_list.append(unitValue)

#Now lets fill the knapsack, intake is how much is in the bag at the moment!

intake = 0

max\_value = 0

factor = True

while(factor):

max\_index = unitValues\_list.index(max(unitValues\_list, default=0))

# this gives the index of the max valued element

if(weight\_list[max\_index] <= capacity):

# In this case, full item is taken in

intake = weight\_list[max\_index]

capacity -= weight\_list[max\_index]

max\_value += value\_list[max\_index]

#©Jaspreet Singh 18CSU094

else:

# weight\_list[max\_index] > capacity

# In this case, fraction to be taken

fraction = capacity / weight\_list[max\_index]

max\_value += value\_list[max\_index]\*fraction

capacity = int(capacity - (weight\_list[max\_index] \* fraction))

weight\_list.pop(max\_index)

value\_list.pop(max\_index)

unitValues\_list.pop(max\_index)

print(weight\_list)

n -= 1 #no. of items left

factor = ((n != 0) if ((capacity != 0) if True else False) else False)

return max\_value

if \_\_name\_\_ == "\_\_main\_\_":

value\_list = []

weight\_list = []

#The first line of the input contains the number 𝑛 of items and the capacity 𝑊 of a knapsack.

#The next 𝑛 lines define the values and weights of the items.

n , capacity = map(int, input('n, capacity: ').split())

for i in range (n):

value , weight = map(int, input('value, weight: ').split())

value\_list.append(value)

weight\_list.append(weight)

#Output the maximal value of fractions of items that fit into the knapsack.

print("{:.10f}".format(knapsack(n, capacity, value\_list, weight\_list)))

**Code with Output:**

A screenshot of a cell phone

Description automatically generated