CSE 204: Data Structures and Algorithms I Sessional

Online: Dynamic Programming (B1)

0-1 Knapsack Problem with Repetition

We will now consider the 0-1 knapsack problem with repetition i.e. we are allowed to pick an item as many times as we want. The thief again has a bag (or “knapsack”) of capacity *W* kg. There are *n* items to pick from, of weights $w_1,...,w_n$ and values $v_1,...,v_n$. You need to implement an algorithm to find the most valuable combination of items the burglar can fit into his bag if the items can be repeated again and again, but taking fractions of the items is not allowed.

Input format:

The first line will contain an integer *n* denoting the number of items. The next *n* lines will each contain 2 numbers separated by space where the first number denotes the weight of the corresponding item and the second its value (both will be integers). Finally, the last line will contain an integer W denoting the total weight of the bag.

Input will be given in a file named **input.txt** .

Example: input.txt

```
5
4 12
2 1
1 1
10 4
2 3
15
```

Output format:

The total value of items taken.

Example: output.txt

```
40
```

In this case, item 1 is picked 3 times and items 3,5 are picked once each.