# Fractional Knapsack Problem (FKP)



- Let $S$ be a set of 'n' items, where each item $i$ has a positive benefit $b_i$ and a positive weight $w_i$.

- Goal - find the maximum benefit subset that does not exceed total weight $W$.

- In FKP, we are allowed to take arbitrary fraction, $x_i$ of each item, i.e the solution is a set of values $x_i$ such that

$$0 \le x_i \le w_i \quad \text{for all } i \text{ and}$$

$$\sum_{i \in S} x_i \le W$$

- The total benefit of the items taken is determined by the sum.

$$\sum_{i \in S} b_i (x_i / w_i)$$

- The general method for the FKP is to compute the value index for each item $i$

$$v_i = b_i / w_i$$

- Then we select items to include in the knapsack, starting with the highest value index.

Fractional Knapsack (S, W)

```
for i = 1 to |S| do
    x_i = 0
    v_i = b_i / w_i
    Insert (v_i, i) into a heap H (max value index at root)
w <- 0
While w < W do
    Remove the max value from H.
    a = min{ w_i , W - w }
    x_i = a
    w = w + a
```

(AJAY RAWAT)

## Analysis

- Assuming $S$ is a heap based priority queue and then the removal has complexity $\Theta(\log n)$, So to remove 'n' elements takes $O(n \log n)$

- If we use a Circular list for $S$, the removal is $O(1)$ So the algorithm is $O(N)$, including the Sort we again have $O(n \log n)$

## Example

Consider 5 items along their respective weights and values

$I = \langle I_1, I_2, I_3, I_4, I_5 \rangle$
$w = \langle 5, 10, 20, 30, 40 \rangle$
$v = \langle 30, 20, 100, 90, 160 \rangle$

Capacity of knapsack $W = 60$, find the Solution to the FKP

**Solution**

| item | $w_i$ | $v_i$ |
|---|---|---|
| $I_1$ | 5 | 30 |
| $I_2$ | 10 | 20 |
| $I_3$ | 20 | 100 |
| $I_4$ | 30 | 90 |
| $I_5$ | 40 | 160 |

Taking value per weight ratio i.e $p_i = v_i / w_i$

| Item | $w_i$ | $v_i$ | $p_i = v_i/w_i$ |
|---|---|---|---|
| $I_1$ | 5 | 30 | 6.0 |
| $I_2$ | 10 | 20 | 2.0 |
| $I_3$ | 20 | 100 | 5.0 |
| $I_4$ | 30 | 90 | 3.0 |
| $I_5$ | 40 | 160 | 4.0 |

Now arrange the value of $p_i$ in decreasing order

| Item | $w_i$ | $v_i$ | $p_i = v_i / w_i$ |
|---|---|---|---|
| $I_1$ | 5 | 30 | 6.0 |
| $I_3$ | 20 | 100 | 5.0 |
| $I_5$ | 40 | 160 | 4.0 |
| $I_4$ | 30 | 90 | 3.0 |
| $I_2$ | 10 | 20 | 2.0 |

Now fill the knapsack according to the decreasing value of $p_i$

| Weight | | Value |
|---|---|---|
| 35 | 60 | $140 \rightarrow \left(\frac{160 \times 35}{40}\right)$ |
| 20 | | 100 |
| 5 | | 30 |

weight  value  $(30 + 100 + 140 = \underline{270})$

## To Show 0/1 Knapsack Cannot be Solved by Greedy

Three items

| | item1 | item2 | item3 |
|---|---|---|---|
| Value | \$60 | \$100 | \$120 |
| Weight | 10 | 20 | 30 |
| Per unit | 6\$/p | 5\$/p | 4\$/p |

Knapsack 50 weight

With Greedy method

| | |
|---|---|
| \$100 | 20 |
| \$60 | 10 |

\$160

or

| | |
|---|---|
| 30 | \$120 |
| 10 | \$60 |

\$180

optimal solution

| | |
|---|---|
| 20/30 | \$80 |
| 20 | \$100 |
| 10 | \$60 |

= 240\$