The Fractional Knapsack Problem

- Given: A set S of n items, with each item i having
  - p_i - a positive profit
  - w_i - a positive weight
- Goal: Choose items, allowing fractional amounts(x_i), to maximize total profit but with weight at most m.

$$\text{maximize} \sum_{1 \le i \le n} p_i x_i$$

$$\text{subjected to} \sum_{1 \le i \le n} w_i x_i \le m$$

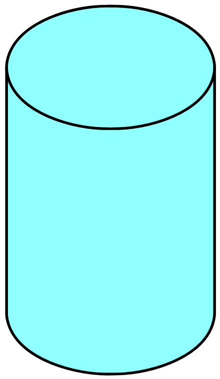
$$\text{and } 0 \le xi \le 1, \qquad 1 \le i \le n$$

Example-4.4: Sahni (Page-218)

Greedy decision property:-

Select items in decreasing order of profit/weight.

Items:	1	2	3
w_i :	18	15	10
p_i :	25	24	15
Value: (p_i / w_i)	1.39	1.6	1.5



Solution:

- 15 of i_2
- 5 of i_3
- 0 of i_1

Knapsack = 20

- Solution vector

  $(x_1,x_2,x_3) = (0,1,1/2)$

- Profit $= 25*0 + 24*1 + 15*1/2$

  $= 0+24+7.5$

  $=31.5$

Algorithm-4.3: Sahni (Page-220)

Greedy algorithm for the fractional Knapsack problem

```
Algorithm GreedyKnapsack(m,n)
//P[1:n]  and w[1:n] contain the profits and weights
// respectively of the n objects ordered such that
//p[i]/w[i]>=p[i+1]/w[i+1].
//m is the knapsack size and x[1:n] is the solution
// Vector.
{
        for i=1 to n do  x[i]=0;   // Initialize x.
        U=m;
        for i=1 to n do
        {
                if ( w[i]>U ) then break;
                 x[i]=1;  U=U-w[i];
        }
        if ( i <=n) then x[i]= U/w[i];
}
```

Thank You

Stay Safe