Welcome ☺

Agenda : Knapsack problems
variations of Knapsack.

Knapsack Problem $\Longrightarrow$ Given N objects with their profit / loss value v[i] & weight w[i]

A bag is given with capacity W that can be used to carry objects s.t

total sum of selected objects $\leq$ W &

sum of profit / loss is max / min

Type 1 Fractional Knapsack (objects can be divided)

Q Given N cakes with their happiness and weight Find max. total happiness that can be kept in a bag with capacity W. (cakes can be divided)

eg: N=5

W=40

$h = [\ 3\ \ \textcircled{8}\ \ \textcircled{10}\ \ 2\ \ \textcircled{5}\ ]$ ← ✗ greedy w.r.t h[] does not work all the time.

$w = [\ 10\ \ 4\ \ 20\ \ 8\ \ 15\ ]$ ✓ ✓ ✓

$\frac{3}{10} = 0.3$ ✓

$4 + 20 + 15 = 39$

$40 - 39 = 1$

$\frac{2}{8} = 0.25$

Ans = 8+10+5

= 23 + 0.3

= 23.3

eg: N=5

W=40

$h = [\ 3\ \ 8\ \ 10\ \ 2\ \ 5\ ]$ ← greedy w.r.t h[]
if w.t ∀ is equal

$w = [\ 10\ \ 4\ \ 40\ \ 8\ \ 15\ ]$

parts = 10 4 40 8 15

h/w = 0.3 2 0.25 0.25 0.33 ✓ greedy

$2*4 + 0.33*15 + 0.3*10 + 0.25*11 = \underline{\underline{18.75}}$

Ans

Sol 1

1) Sort w.r.t. h[i]/wt[i]

2) Select cakes/parts in descending order of h[i]/wt[i]

T.C → N Log N   S.C → O(N)

Type 2 $\boxed{0-1}$ Knapsack (objects cannot be divided)

$0-1$ → select or not select

Q Given N toys with their happiness & weight → cost
Find max. total happiness that can be kept in a bag with capacity W

eg: $N = 4$ $\quad H[\ \overset{\checkmark}{4}\quad 1\quad \overset{\checkmark}{5}\quad 7\ ]$

$W = 7$ $\quad W[\ 3\quad \underset{\checkmark}{2}\quad 4\quad \underset{\checkmark}{5}\ ]$

$h/w \quad 1.33 \quad 0.5 \quad 1.25 \quad 1.4$

greedy app.

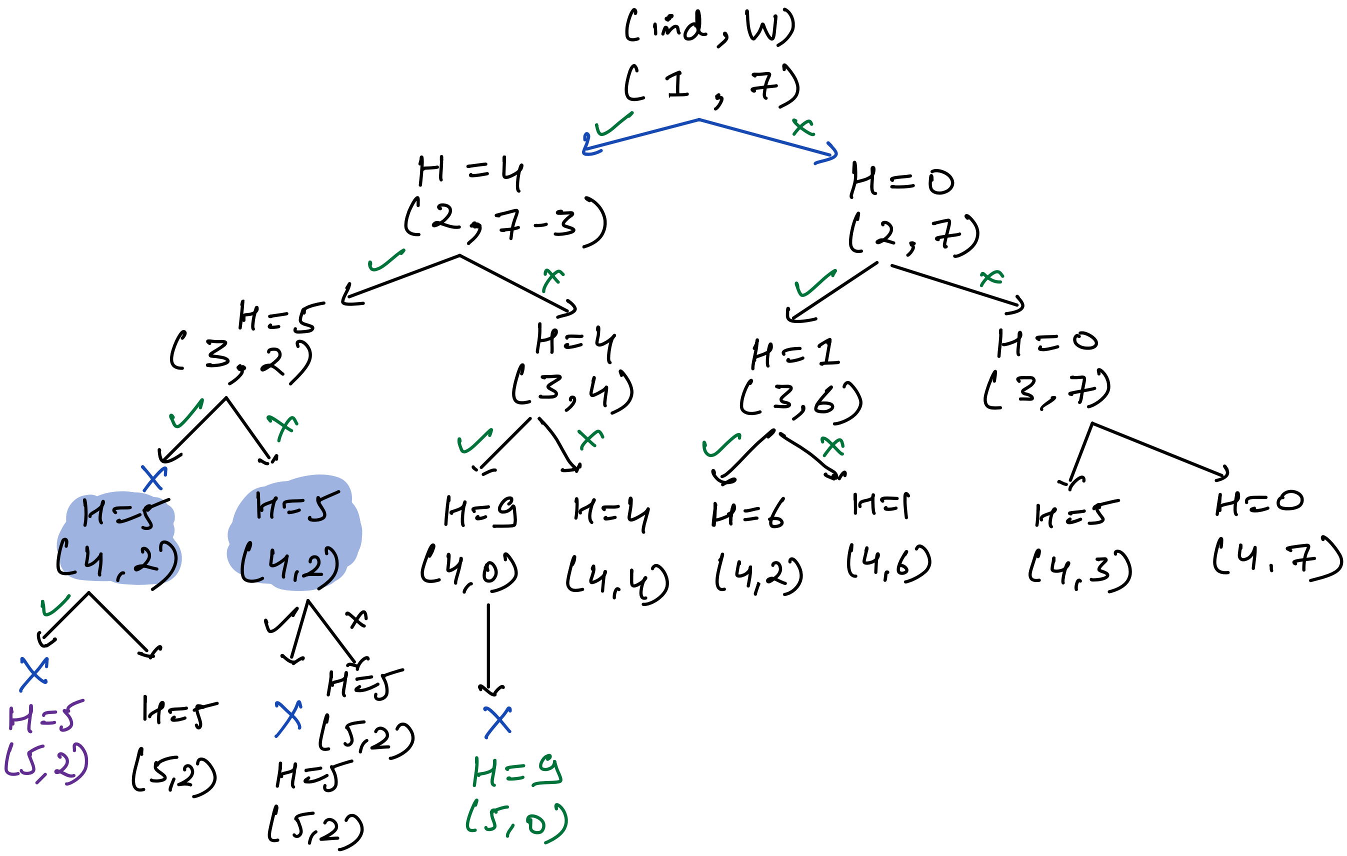
Ans $= 7 + 1 = 8$ ✗ wrong answer.

Ans $= 4 + 5 = \underline{\underline{9}}$.

Brute force ∀ subsets of toys, check if $\sum wt \le W$,
take max $\sum h[i]$ T.C ⇒ $O(2^N)$

$N = 4 \quad H[\ \overset{1}{4}\quad \overset{2}{1}\quad \overset{3}{5}\quad \overset{4}{7}\ ]$

$W = 7 \quad W[\ 3\quad 2\quad 4\quad 5\ ]$



$$state \longrightarrow (\overset{1 \to N}{index}, \overset{0 \to W}{capacity})$$

$$\#\ unique\ states \Rightarrow N * (W+1) \simeq N*W < 2^N$$

ans[i][j] ⇒ Max. happiness considering first i objects & capacity j.

$$ans[i][j] \begin{cases} \xrightarrow{\checkmark} h[i] + ans[i-1][j-w[i]] \\ \xrightarrow{\times} ans[i-1][j] \end{cases}$$

$$if\ (i==0\ ||\ \underset{\substack{\downarrow \\ \text{no weight left.}}}{j==0}) \rightarrow ans[i][j] = 0$$

Code

```
∀ i,j   ans[i][j] = 0
for i -> 1 to N   // 1 based index
{
    for j -> 1 to W
    {
        if ( j < w[i] )  ans[i][j] = ans[i-1][j]
        else  ans[i][j] = max( h[i] + ans[i-1][j-w[i]] ,
                               ans[i-1][j] )
    }
}
return ans[N][W]
```

cannot select i^{th} object b/w weight > remaining capacity

select: h[i] + ans[i-1][j-w[i]]

Not select: ans[i-1][j]

T.C → O(N * W)

S.C → O(N * W)

optimize

only store 2 rows.

S.C ⇒ O(2 * W) ≃ O(W)

Type 3 Unbounded Knapsack | 0-N Knapsack

(objects cannot be divided)

(one object can be selected multiple times)

Q Given N toys with their happiness & weight (cost). Find max. total happiness that can be kept in a bag with capacity W. A toy can be selected multiple times.

eg:

N = 3 h = [2 3 5]

W = 8 w = [3 4 7]

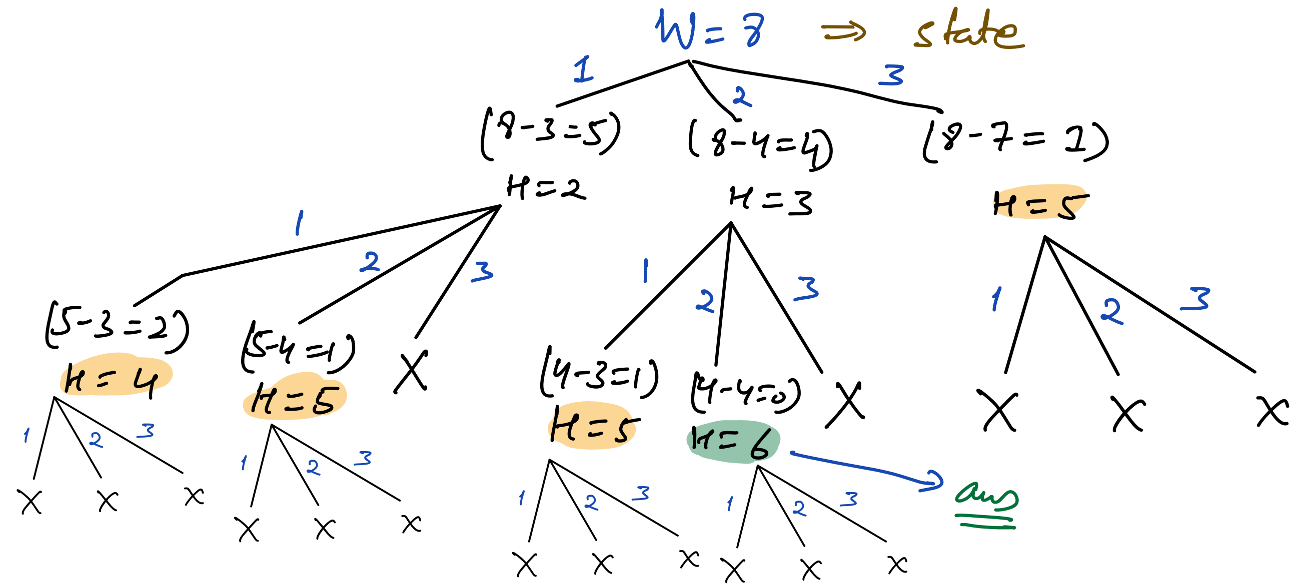
select 2 times.

Ans = 3 * 2 = 6

Brute force

→ Select any toy & place it in the bag. at every step till capacity is available.

eg: N = 3 h = [2 3 5]
W = 8 w = [3 4 7]



\# unique states. = W

height = W
T.C ⇒ $O(N^W)$

weight

ans[i] ⇒ max. happiness with capacity i

toys.

ans[i] ⇒ $\forall_j$ max (h[j] + ans[i − w[j]]

T.C → O(N*W)
S.C → O(W)

Code

```
∀i, ans[i] = 0
for ( i → 1 to W ) → weights
{
    for ( j → 1 to N ) → toy
    {
        if ( i ≥ w[j] )
            ans[i] = max ( ans[i] , h[j] + ans[ i − w[j] ] )
    }
}
return ans[W]
```