Welcome ☺

Agenda : DP 3
Knapsack problems
Variations of Knapsack

Knapsack problem → Given N objects with their profit / loss.
value [i] & weight [i]
A bag is given with capacity W that can be used to carry objects s.t
1) total sum of selected objects ≤ W
2) Sum of profit / loss is max / min.

Type 1 Fractional Knapsack.

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

eg: N = 5
W = 40

	1	2 ✓	3 ✓	4	5 ✓
h	3	8	10	2	5
w	10	4	20	8	15

greediness w.r.t h[] will not work.

total happiness = 23
weight = 39

Ans ⇒ 23.3

	1	2	3	4	5
$N = 5$ h	3	8	10	2	5
$W = 40$ w	10	4	20	8	15
H/w	$\frac{3}{10} = 0.3$	2	0.5	0.25	0.33

Sol 2 ⇒ 1) Sort w.r.t H/w

2) Select cakes/part in descending order of $h[i]/w[i]$

TC ⇒ $O(N \log N)$ SC ⇒ $O(N)$

Type 2 0-1 Knapsack. (objects cannot be divided)

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

	✓	✓	✓	✓
$N = 4$ H	4	1	5	7
$W = 7$ w	3	2	4	5
H/w	1.3	0.5	1.25	1.4

ans = 5 + 4 = 9
weight = 4 + 3 = 7

greedy app.

ans = 7 + 1 = 8
weight ⇒ 5 + 2 = 7

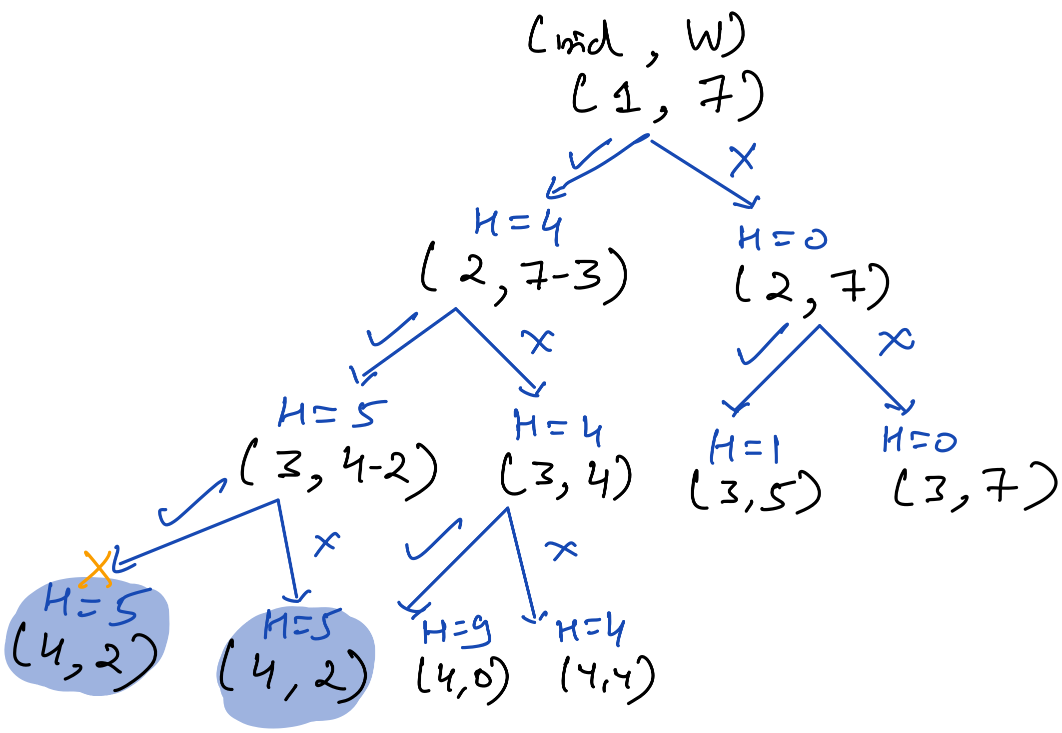
DOES NOT WORK.

Brute force $\forall_{subset}$ of toys, check if $\sum wt \le W$,

take max $\sum h[i]$

T.C $= O(N * 2^N) \rightarrow O(2^N)$

		1	2	3	4
$N=4$	H [	4	1	5	7]
$W=7$	w [	3	2	4	5]



* **State** $\Rightarrow$ (index, capacity), with index $1 \rightarrow N$ and capacity $0 \rightarrow W$

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

ans[i][j] $\Rightarrow$ Max. happiness considering first i objects & capacity j.

ans[i][j] $\Rightarrow$
- ✓ $\rightarrow$ h[i] + ans[i-1][j - w[i]]
- ✗ $\rightarrow$ ans[i-1][j]

if (i == 0 || j == 0) ans[i][j] = 0

Code

```
∀ i,j   ans[i][j] = 0
for( i -> 1 to N)
{
    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]
```

T.C ⇒ $O(N*W)$

S.C ⇒ $O(N*W)$

↓ optimize

Use 2 1D arrays

S.C ⇒ $O(W)$

$\boxed{10:45}$

N = 4   W = 7

	1	2	3	4
H	4	1	5	7
W	3	2	4	5

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	4	4	4	4	4
2	0	0	1					
3	0							
4	0							

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 and weight (cost). Find max. total happiness that can be kept in a bag with capacity W. A toy can be selected multiple times.

eg:

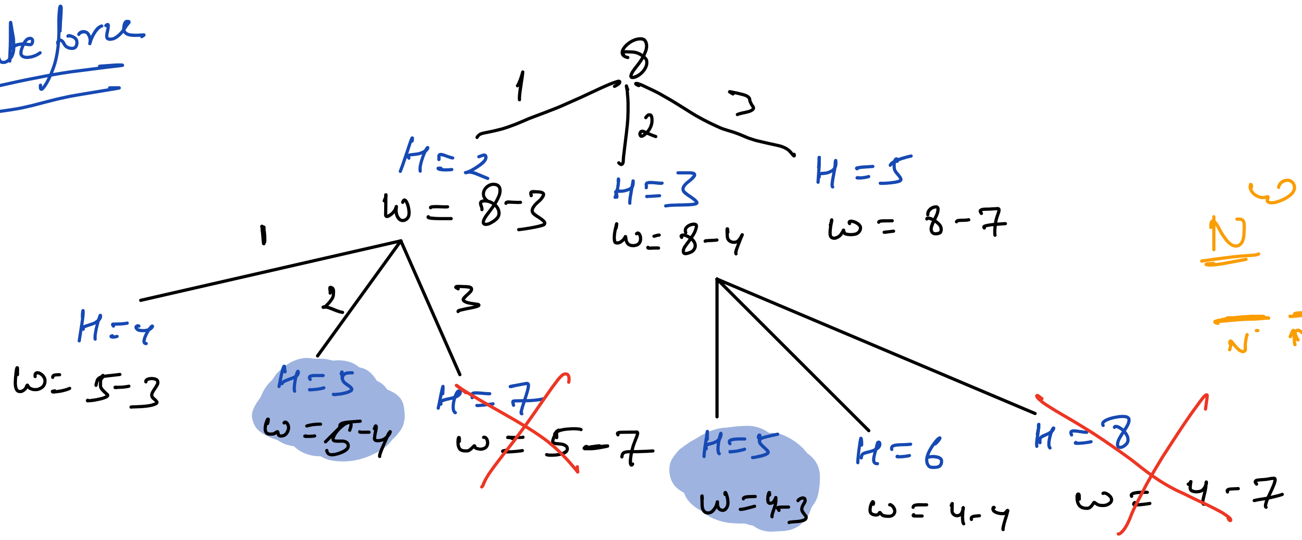
$N = 3 \quad h = [\,2 \quad 3 \quad 5\,]$

$W = 8 \quad w = [\,3 \quad 4 \quad 7\,]$

happiness = 6

selected 2 times

Brute force



N^W

$\overline{N} \cdot \overline{N} \; \overline{N} \; \overline{N}$

state ⇒ remaining capacity

\# unique ⇒ W

height = W

T.C = $O(N^W)$

ans[i] $\Rightarrow$ max. happiness with capacity i

ans[i] $\Rightarrow$ $\forall_j$ max(h[j] + ans[i - w[j]])

Code

```
                  W
∀i   ans[i] = 0

for( i -> 1 to W)
{
    for( j -> 1 to N)
    {
        if ( i >= w[j] )
        {
            ans[i] = max( ans[i], h[j] + ans[ i - w[j] ])
        }
    }
}
return ans[W]
```

T.C $\Rightarrow$ $O(N*W)$

S.C $\Rightarrow$ $O(W)$