Welcome ☺

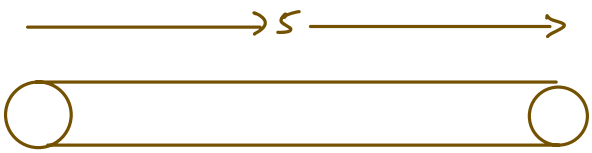
Agenda : 4 problems.

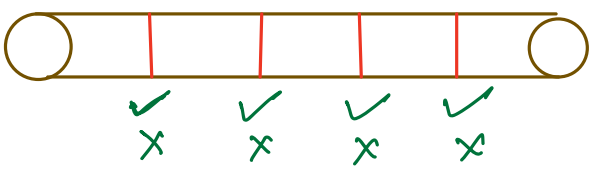
Q Given a rod of length N, and an array A of length N
A[i] ⇒ price of i length rod. (index is 1 based)
Find the max value that can be obtained by cutting rod into some pieces & selling them.

N = 5

A = [1 4 2 5 6]

1 2 3 4 5





#ways = 2^{N-1} = 16

sold length	Total price.
5	6
1+4	6
3+2	6
3+1+1	2+1+1 = 4
2+2+1	4+4+1 = 9 (Ans)
2+1+1+1	4+1+1+1 = 7
1+1+1+1+1	5

Unbounded Knapsack.

W ⟶ length of rod = N

wt[i] ⟶ length of piece of rod.

value[i] → A[i]

dp[i] ⇒ max. value if capacity is i^{N}

∀i dp[i] = 0

```
for ( i → 1 to N )  // capacity.
{
    for ( j → 1 to i )  // cuts of rods / objects
    {
        dp[i] = max ( dp[i] , A[j] + dp[i-j] )
    }
} return dp[N]
```

T.C → $O(N^2)$

S.C → $O(N)$

Q In how many ways can we make N by using coins. given in the array i.e

A[i] → value of i^{th} coin

One coin can be selected multiple times.

A: Ordered selection. (x, y) ≠ (y, x)

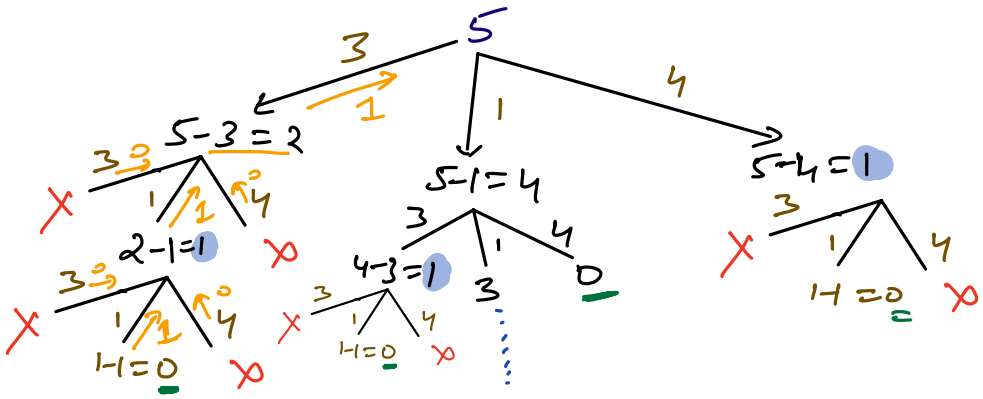
N = 5

A = [3 1 4]

{1, 4} {3, 1, 1} {1, 1, 3}

{4, 1} {1, 3, 1} {1, 1, 1, 1, 1}

Ans = 6



dp[i] ⇒ # ways to select coins s.t sum = i

```
∀i. dp[i] = 0
dp[0] = 1
for i → 1 to N
{
   for (j → 0 to (A.length() - 1)) // index
   {
      if( A[j] ≤ i )
      {
         dp[i] += dp[ i - A[j] ]
      }
   }
}
return dp[N]
```

T.C → O(N * A.length)

S.C → O(N)

B → UnOrdered selection. (x, y) = (y, x)

N = 5

A = [3 1 4]
1 2 3

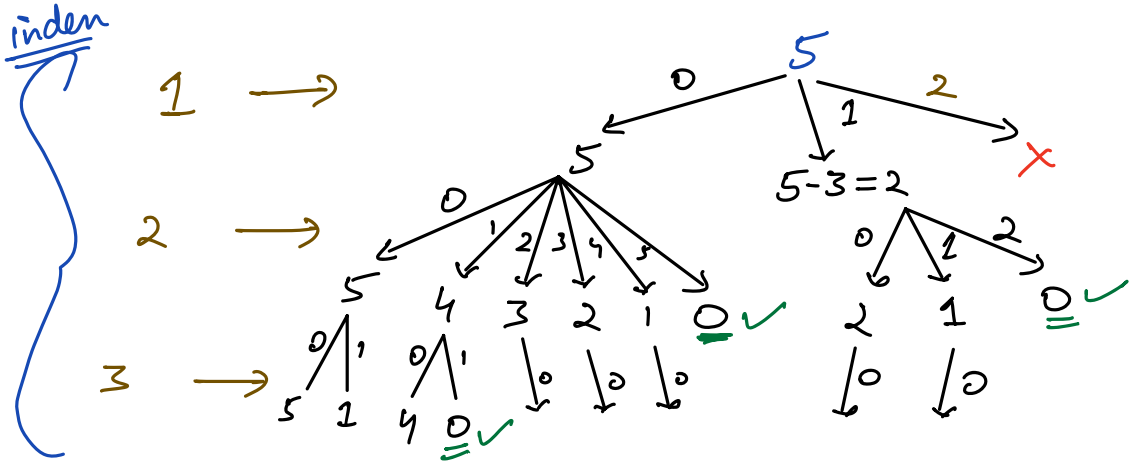
{1,4} same. {4,1}

{3,1,1} {1,3,1} same

{1,1,3} same

{1,1,1,1,1}

Ans = 3



dp[i] ⇒ # ways to select coins s.t sum = i

dp[0] = 1

```
for(j → 0 to (A.length() - 1)) // inden
{
    for i → 1 to N
    {
        if( A[j] ≤ i )
        {
            dp[i] += dp[ i - A[j] ]
        }
    }
}
return dp[N]
```

T.C → O(N * A.length)

S.C → O(N)

Q Given N toys with their happiness & weight (cost)
Find max. total happiness that can be kept in a bag with capacity W (toys cannot be divide)

Constraints

$1 \le N \le 500$

$1 \le h[i] \le 50$

$1 \le wt[i] \le 10^9$

$1 \le W \le 10^9$

0-1 Knapsack → T.C → $O(N * W)$

$500 * 10^9 = 5 * 10^{11}$

TLE

$dp[N][H]$ → max. happiness

$dp[N][H]$ → min. capacity (W) required to achieve happiness H with N elements.

max. happiness = $500 * 50$

= 25000

Code

Min

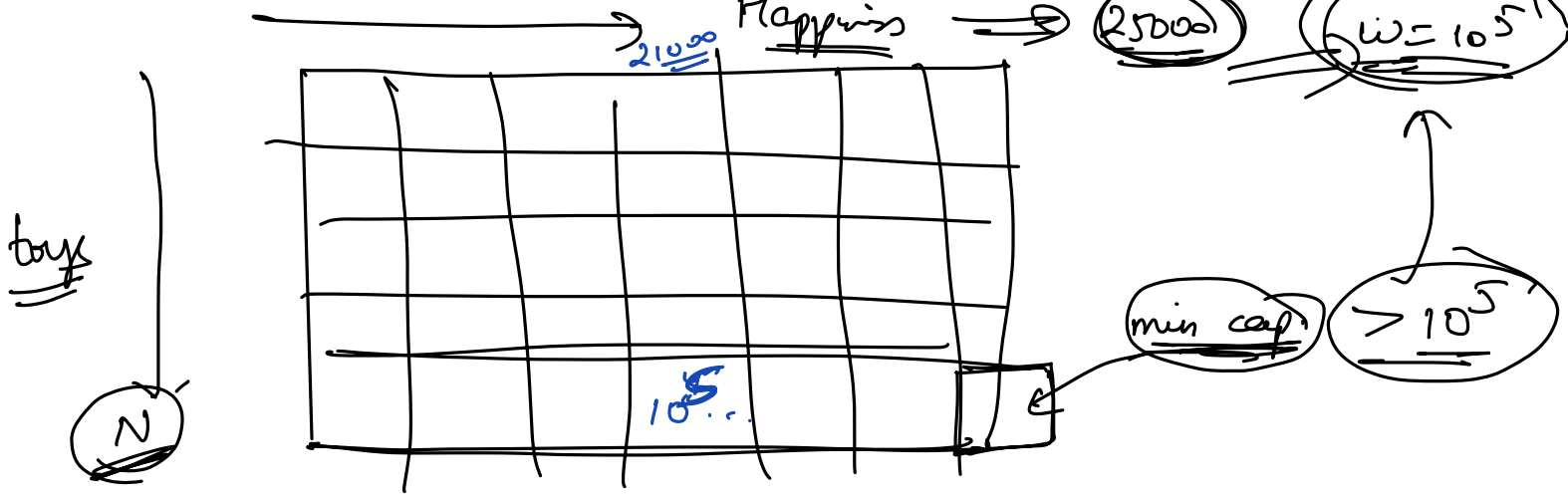
for i → H to 0

if ($dp[N][i] \le W$) return i

T.C ⇒ $N * H$

= $500 * 25000$

= 1.25×10^7 ✓


toys
N
2100
Happiness
25000
10^5
10^5
min cap
$> 10^5$