$\underline{\underline{\text{Q=1}}}$

h_1	h_2	h_3	.	.	.	.	h_n

n stone

$$\hookrightarrow \underbrace{f(i)}_{} = \min\Big(\underbrace{|a_{i-1} - a_i| + f(i-1)}_{\text{one choice}} \;,\; \underbrace{|a_{i-2} - a_i| + f(i-2)}_{\text{second choice}}\Big)$$

which returns the minimum cost to reach from stone 1 to stone i

[dp[i]] →

Q => 2

$$f(i) = \min\left(|a_i - a_{i-j}| + f(i-j) \right)$$

min cost to reach at i from first stone

$$\forall\ j \in (i-k,\ i-1)$$

Q :) 3

$$f(a, i) = \max \begin{cases} h_{i0} + \max(f(1, i-1), f(2, i-1)) \\ h_{i1} + \max(f(0, i-1), f(2, i-1)) \\ h_{i2} + \max(f(0, i-1), f(1, i-1)) \end{cases}$$

0th activity on day i

$f(a, i)$ → max happiness till the i^{th} day

$f(i) \leftarrow \max$

a → 0 → sum
a → 1 → buys
a → 2 → hw

$\boxed{\text{dimension of dp}} \rightarrow 2d$

Q.4

$i_1 \quad i_2 \quad i_3 \; . \; - - - - - - \; i_j \; - - - - - - \; i_n$

$\{1, 2, 3\}$

$\downarrow$

2^n ways

$\{ \{1\}, \{3\}, \{1,3\}, \{1,2,3\}, \{2\}, \{1,2\}, \{2,3\}, \{\} \}$

Target wt → W

$f(i, TW)$ =
$$\begin{cases} f(i-1, TW) & \text{if } wt[i] > TW \text{ (Not pick)} \\ \max\left(\underbrace{f(i-1, TW)}_{\text{Not pick}},\ \underbrace{val[i] + f(i-1, TW - wt[i])}_{\text{pick}}\right) & \text{if } wt[i] \le TW \text{ (Pick or not pick)} \end{cases}$$

↓

max profit/value we can get till the i^{th} item with target wt → TW

dp[i][j] = if (wt[i] > j) {
dp[i-1][j]
} else {
max(dp[i-1][j], val[i] + dp[i-1][j-wt[i]])
}

max profit till i^{th} element with J as the wt of knapsack

2d array → make 2 one-d arrays

n × W

O(W)

W=8

3 , 30
4 , 50
5 , 60

items

none of them is picked

TW

O(n×w)

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	30	30	30	30	30	30
2	0	0	0	30	50	50	50	80	80
3	0	0	0	30	50	60	60	60	90

90

dp[i-1][j] not pick

dp[i][j] → max profit

val[i] + dp[i-1][j-wt[i]] pick

30 + 0

50 + dp[1][7-4]

50 + 30 = 80

Q::5

$10^3 \times 10^2 \to 10^5$
10^2 $\Big] \to 10^7$

$f(i, v)$ = $\begin{cases} f(i-1, v) & \text{if } val[i] > v \\ \min\big(f(i-1, v), \; wt[i] + f(i-1, v - val[i])\big) & \text{else} \end{cases}$

→ Not pick

pick

that gives the min wt reqd so that the value of selected items is v. till the i^th element ans → for last row → $i = n-1$

max value of the column j such that

$dp[n-1][j] <=$ Knapsack wt

max j → ans

wt	val
3	3
4	5
5	6

$W = 8$

$8 - 5 = 3$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0														
1	0	∞	∞	3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	0	∞	∞	3	∞	4	∞	∞	7	∞	∞	∞	∞	∞	∞
3	0	∞	∞	3	∞	4	5	∞	∞	8	∞	9	∞	∞	12

$W = 8$

full array is filled with $+\infty$

$\infty, \; 4 + 6$

$\infty, \; 4 + \infty \rightarrow \infty$

$\infty, \; 4 + 3$