0-1 Knapsack

Q: There is a robber who wants to rob some houses. In each house there is a gold of weight wt[i] & cost c[i]. Robber has a knapsack which can inculcate max W weight items inside it. The only constraint is, if robber loots a house he will either pick all gold from that house or pick nothing. What is the max value loot he can do.

W = 5 {6, 10, 12}

{1, 2, 3}

ans → 22

Explore all possibility

Recurrence Relation

wt → weight array

c → cost array

$f(i, w)$ → function, gives max profit till the i^{th} house with W wt remaining in knapsack

$$
= \begin{cases} \max\big(f(i+1, w - wt[i]) + c[i],\ f(i+1, w) \big) & \to wt[i] \le W \\ f(i+1, w) & wt[i] > W \end{cases}
$$

2 choices pick or not pick ↑

wt[i] ≤ W → wt if gold at ay hour

wt[i] > W → we only have one choice

To uniquely identify a state we require 2 parameters. → max space required → 2 dimensional

$W = 5$ $\{6, 10, 12\}$ cost

$3 - 2 = 1$

$4 - 2 = 2$

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

$O(n \times w) \rightarrow TC$

$SC \rightarrow O(n \times w)$

prev

curr

$2 \rightarrow 1D$ arr

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	6	6	6	6	6
2	0	6	10	16	16	16
3	0	6	10	16	18	22

12 + 16, 16

22, 16

$O(w) \rightarrow SC$

$$dp[i][j] = \begin{cases} \max(dp[i-1][j - wt[i]] + c[i],\ dp[i-1][j]) & wt[i] \le j \\ dp[i-1][j] & wt[i] > j \end{cases}$$

$f(i, j)$

$dp[i][j] \rightarrow$ max profit

till the i^{th} element with j as Bag capacity

In DP, Time complexity depends on the no. of unique states.

TC = no. of unique states × (time consumed in each state)

TD-DP

Q:- There are n friends who want to go to party. The restraint is, all of them have 2 choices–

① Either they can go alone $n \le 10^5$

② They can go in a pair.

Calc the no. of ways n people go to party.

$n = 3$

↳ 4

(A) (B) (C) → 1

(A B) (C) → 1

(A) (B C) → 1

(A C) (B) → 1

$\rightarrow$ $f(n) = f(n-1) + (n-1) * f(n-2)$

$f(n)$: no. of ways n friends can go to party

$f(n-1)$: choose to go alone

$(n-1)$: no. of pair available

$f(n-2)$: when you go in pair

A | B C D

either to go alone

or make pair

(A) (B) (C) (D)

(A) (BC) (D)

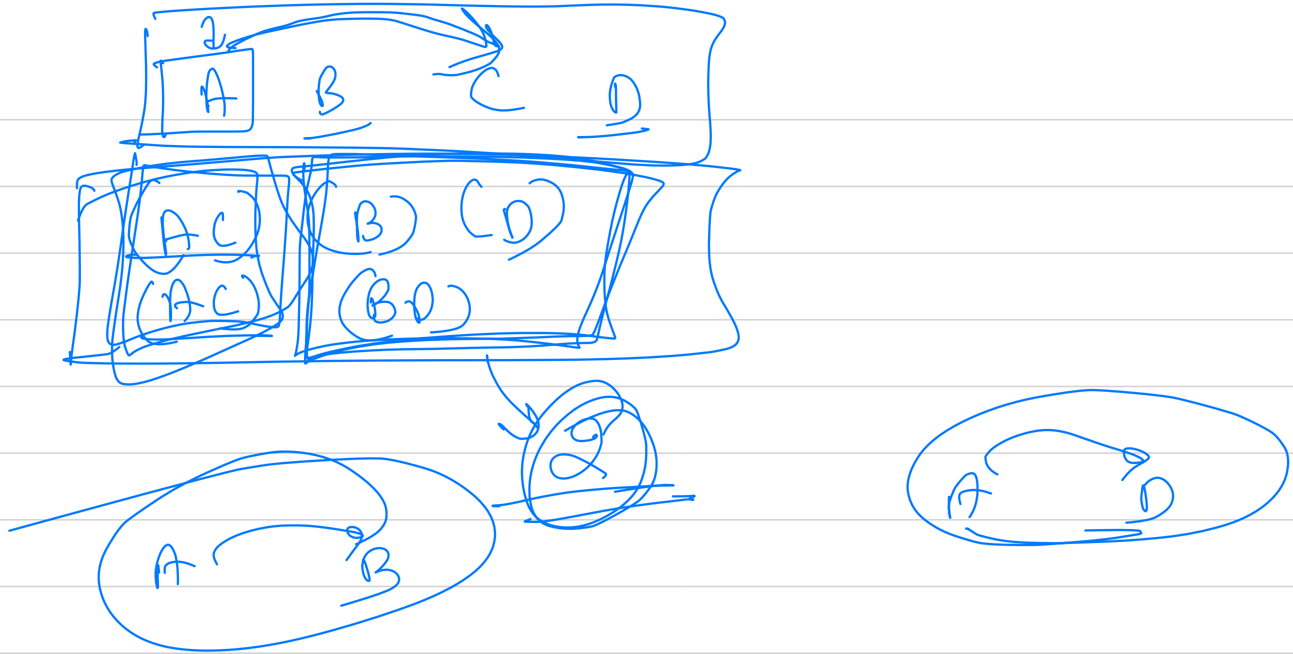
(A) (BD) (C)

(A) (B) (CD)

A B (C D)

(AB) (C) (D)

(AB) (CD)


2
A
B
C
D
(A C)
(B D)
(A C)
(B D)
A
B
A
D

always go alone $\quad \boxed{A} \quad B \quad C \rightarrow n$

(A) (B) (C)

(A) (B C) $\rightarrow n-1$

go alone $\quad A \quad \boxed{B \quad C \quad D} \rightarrow n$

(A) (B) (C) (D)

(A) (B C) (D)

(A) (B) (C D)

(A) (B D) (C) $\rightarrow n-1$

$$f(n) = f(n-1) +$$

A B C D

3 ways

$\boxed{A}$ $\underbrace{B \quad C \quad D \quad E \quad F}$] → (n-1) ways

A B C D → n

(A B) (C) (D)

(A B) (C D)

$n-2$

$(n-1) * f(n-2)$

(A C) (B) (D)

(A C) (B D)

(A D) (B C)

(A D) (B) (C)

$$f(n) = (n-1) * f(n-2)$$

$$f(n) = f(n-1) + (n-1)\,f(n-2)$$

almost fib recurrence

TD

BU