unacademy

# Miscellaneous Topics - Part V

Complete Course on Algorithms - GATE

**Subbarao Lingamgunta** • Lesson 56 • May 27, 2023

# 0/1 knapsack

## using - DP

n = 6   m = 35

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| objects : | ob$_1$ | ob$_2$ | ob$_3$ | ob$_4$ | ob$_5$ | ob$_6$ |
| profit : | 5000 | 80 | 90 | 65 | 150 | 120 |
| weight : | 70 | 300 | 50 | 5<br>~~3~~10 | 100 | 80 |

35, 0   O(n)

01ks(m, n) = The maximum profit we will get with capacity m & no. of objects n.

```
01ks(m, n)      [m = 35, n = 6]
{
    if(m == 0 || n == 0)
        return(0)
    else
        if(w[n] > m)                         BC
            return(01ks(m, n-1))
        else
        {
            a = 01ks(m - w[n], n-1) + p[n]
            b = 01ks(m, n-1)                 WC
            c = max(a, b)                    c
            return(c)
        }
}
```

# Sum of Subsets

i/p: Set of n +ve integers, integer m

o/p: Find any subset whose sum is m.

$$S = \begin{pmatrix} 50 & 80 & 60 & 30 & 25 & 45 & 100 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{pmatrix}$$

$m = 200$

$$SS_1 = (\overset{\checkmark}{50}\ \ \overset{\checkmark}{80}\ \ \overset{\checkmark}{25}\ \ 45) \checkmark$$

$$SS_2 = (30\ \ 25\ \ 45\ \ 100)$$

$200 - 45 = 155$

$155 - 25 = 130$

$130 - 80 = 50$

$50 - 50 = 0$

SOS(m, n) = Find any subset from n-ele set so that its sum is m.

ex

$n = 7 \quad m = 200$

(50, 80, 25, 45)

(30, 25, 45, 100)

$S = (\underset{1}{50}, \underset{2}{80}, \underset{3}{60}, \underset{4}{30}, \underset{5}{25}, \underset{6}{45}, \underset{7}{100})$

$SS = \phi$

$SOS(\overset{m}{200}, \overset{n}{7}) = SOS(m, n-1)$

200, 6

200, 5

200, 4

200, 0 = -1

$B^L$

$O(n)$

$\underset{0,\ 6}{SOS(m, n)} = SS$

$$\underset{200,\ 7}{SOS(m, n)} = \begin{cases} SOS(m - S[n], n-1), \\ \quad SS = SS \cup S[n] \\ SOS(m, n-1) \end{cases}$$

```
SOS(m,n)
{   ss = φ
    if(m == 0)
        return(ss)
    else
    if(S[n] > m)
        return(SOS(m, n-1))
    else{
        SOS(m-S[n], n-1), SS = SS ∪ S[n]
        SOS(m, n-1)
    }
}
```

n

BC

WC

## Without-DP

Time: $2^n$

Space: $O(n)$

## With-DP

SOS(m,n)

m+1

n+1

mn - DFL

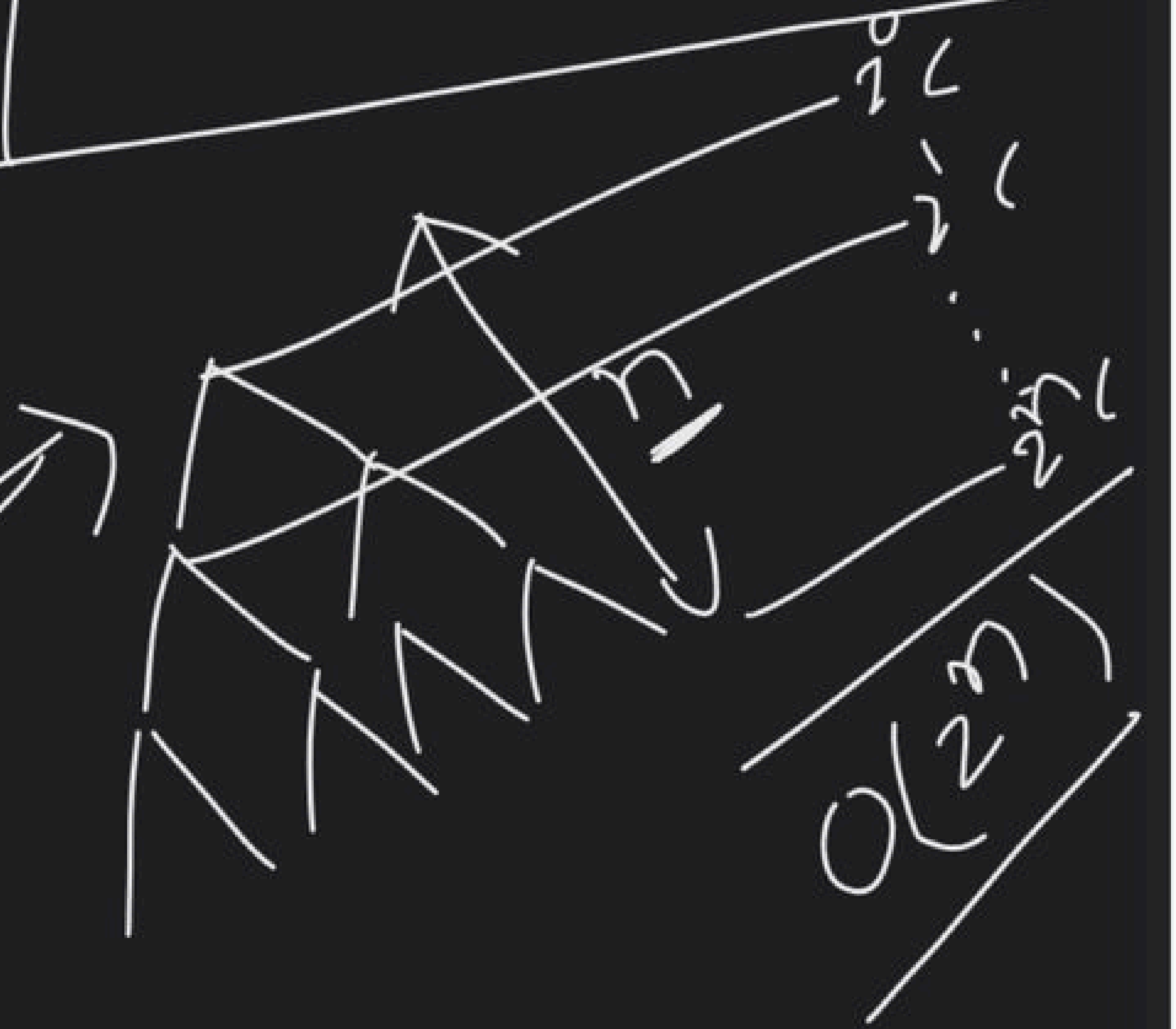
Time O(mn)
WC

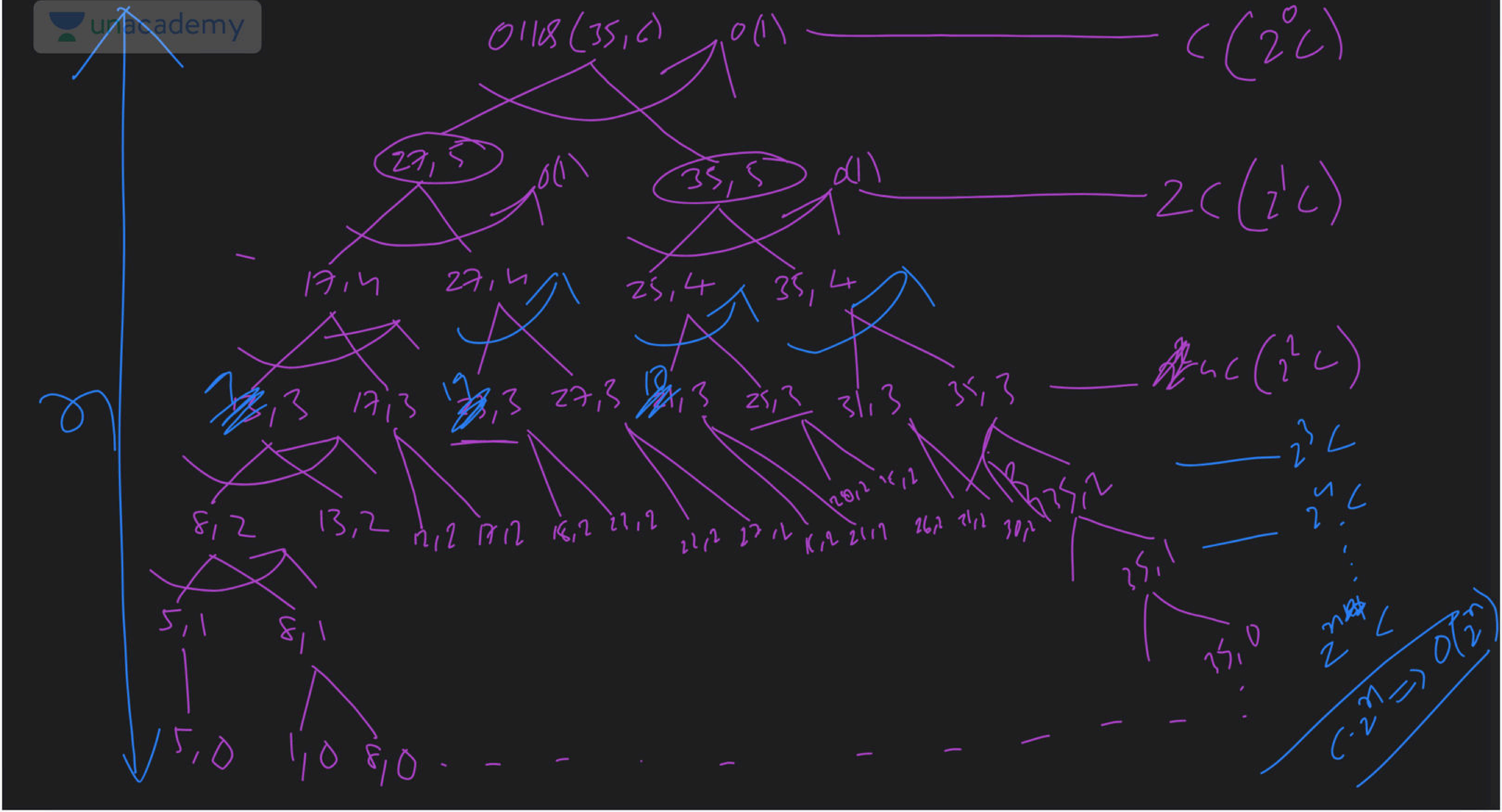
Space = Stack + Table

Stack = n

Table = mn

O(mn)





01KS(35, 6)
O(1)
C (2^0 C)
27, 5
O(1)
35, 5
O(1)
2C (2^1 C)
17, 4
27, 4
25, 4
35, 4
8, 3
17, 3
19
23, 3
27, 3
18
21, 3
25, 3
31, 3
35, 3
4C (2^2 C)
n
8, 2
13, 2
12, 2
17, 2
18, 2
23, 2
22, 2
27, 2
16, 2
21, 2
20, 2
25, 2
26, 2
31, 2
30, 2
35, 2
2^3 C
2^4 C
35, 1
35, 0
5, 1
8, 1
5, 0
1, 0
8, 0
2^n C
C·2^n ⇒ O(2^n)

## Without -DP

Time : $2^n$ (WC)

Space ⇒ extra space
⇓
stack space
⇓
O(n)

## With -DP

01KS(m, n) — DFC ?

| m | n |
|---|---|
| 35 | 6 |
| 34 | 5 |
| 33 | 4 |
| ⋮ | 3 |
| ⋮ | ⋮ |
| 2 | 1 |
| 1 | 0 |
| 0 | |
| 35+1 | 6+1 |

(35+1) (6+1) — DFC

(m+1)(n+1) — DFC

mn — DFC

mn * O(1)

O(mn) Time WC

$O(mn) \cong O(2^n)$

becz of less repeated subp

So it is one of the NPC problem.

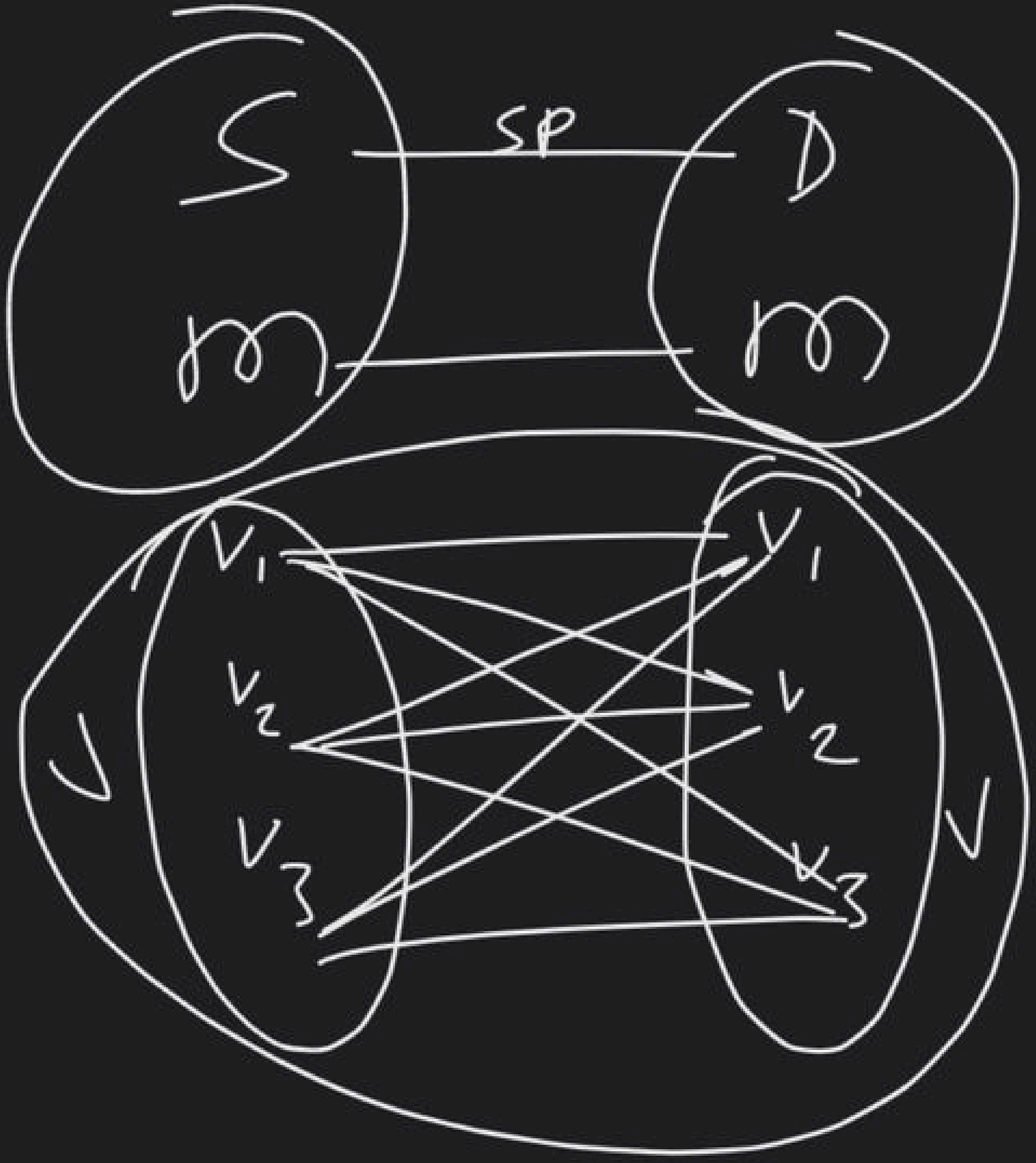
Space complexity
⇓
extra space

stack
⇓
n

info
⇓
DFC
⇓
mn

O(mn)

# All Pairs shortest path



$V^2$ - pairs

Floyd Warshall - Algo

using - DP

⇓

$O(V^3)$ time

$V^2 * V$

APS(i, k, j)

V V V

$V^3$, DP

$= V^3 * O(1)$

Space

⇓

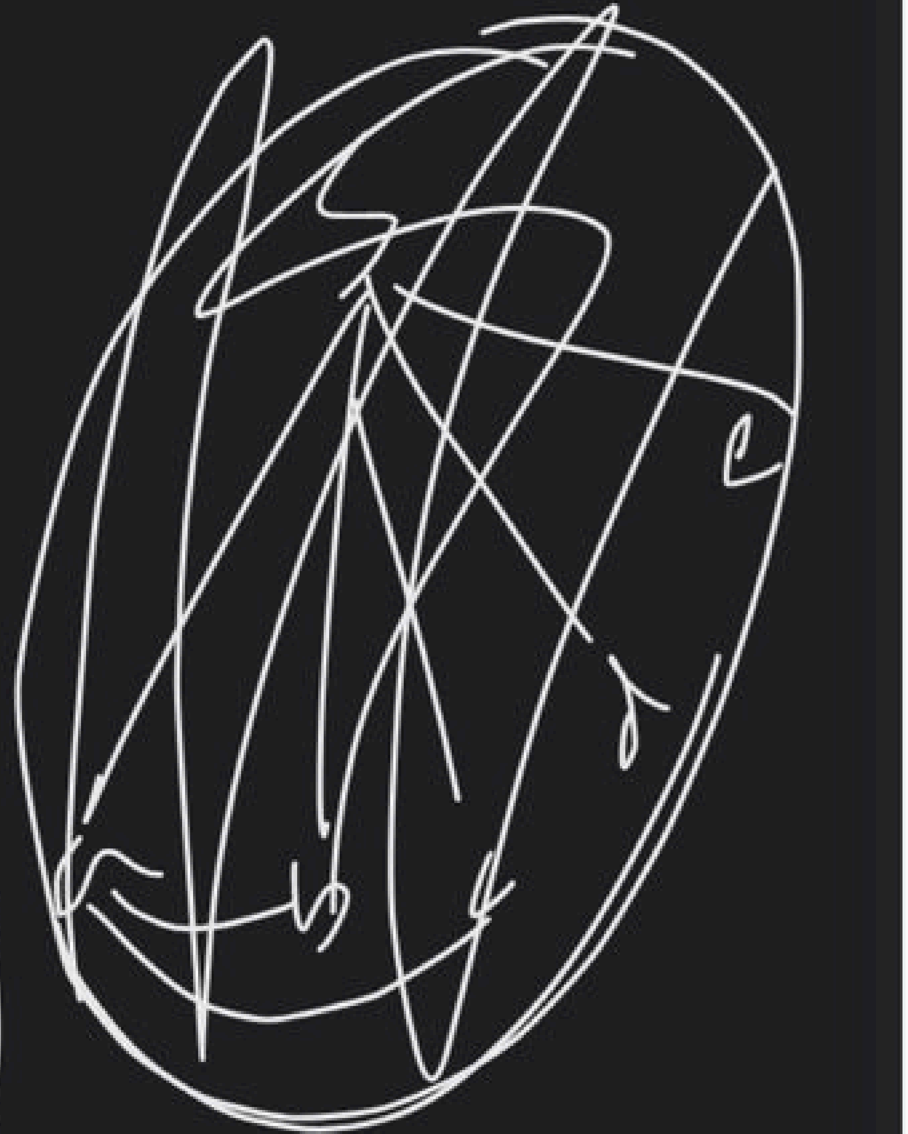
Stack + table

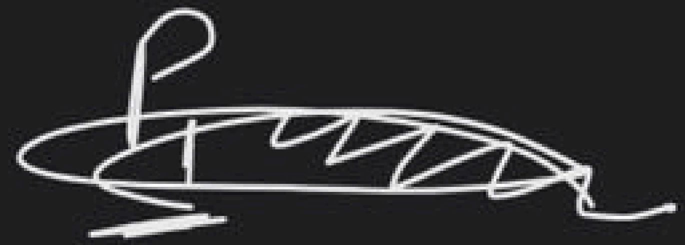
⇓ ⇓

~~V~~ $V^3$

V

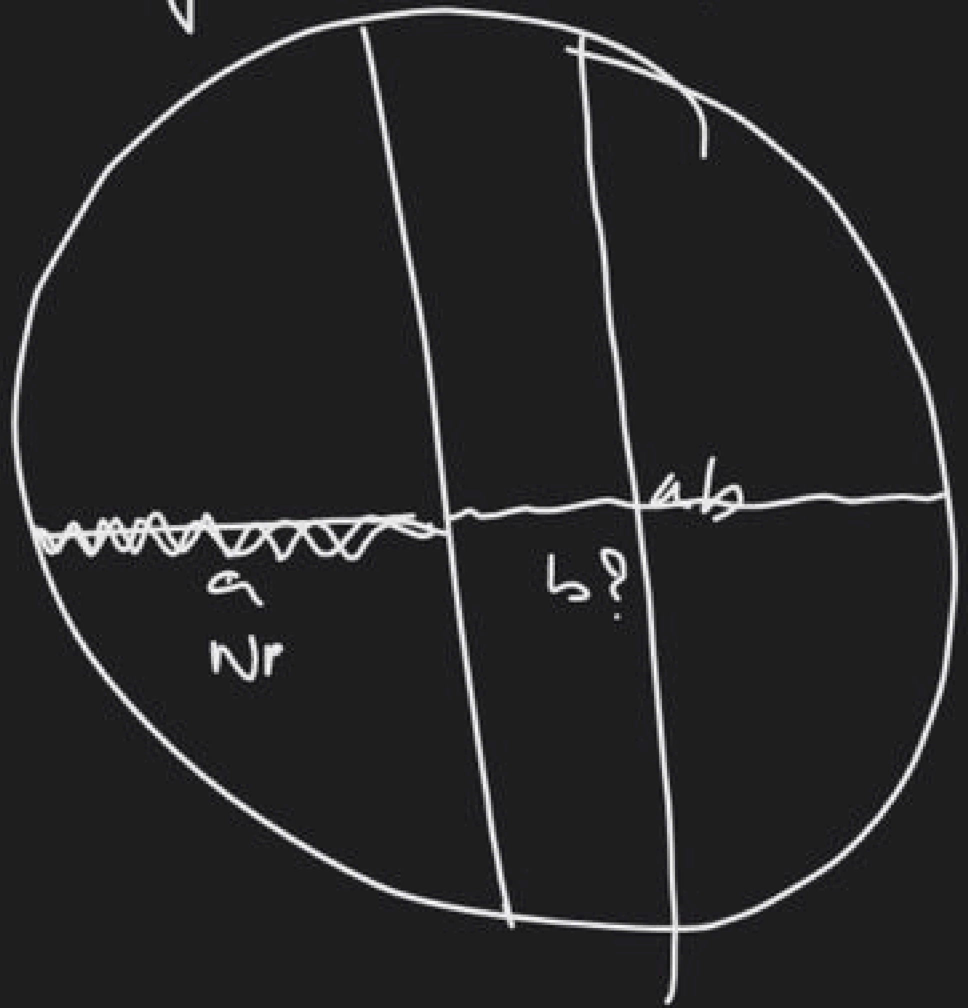
$O(V^3)$





$GC \leq_p SOS \leq_p 01KS \leq_p$

$SI \leq_p$



3-SAT

3-SAT

3-variable Satisfiable problem