

Q You go to a party of N people

2 rules:

1. Either enjoy the party alone.

2. _____ as a pair.

$N \geq 1$

Find the no. of ways of N people to party!

$N=1$

$((1))$

$\rightarrow 1$

$N=2$

$((1)(2)), ((1,2))$

$\rightarrow 2$

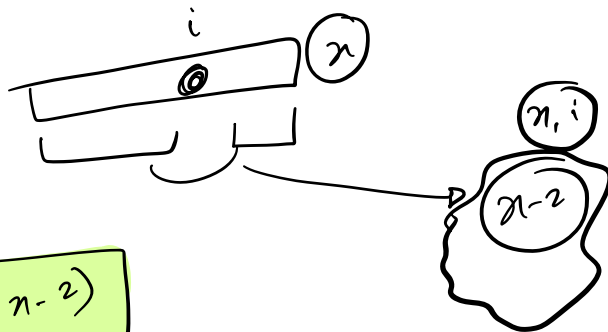
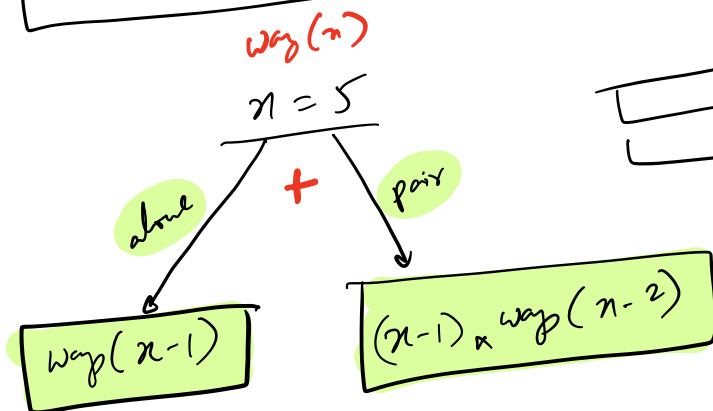
$N=3$

$((1)(2)(3)), ((1,2)(3))$

$\rightarrow 4$

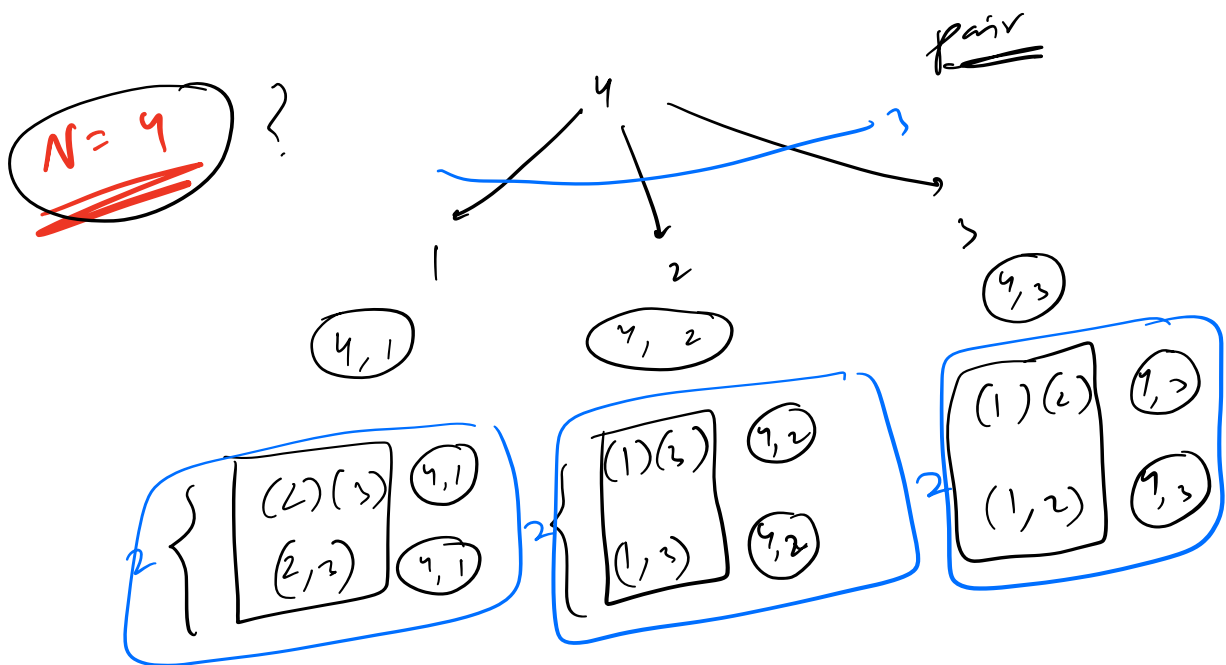
$((1,3)(2)), ((2,3)(1))$

$Way(n) \rightarrow \# \text{ ways of } n \text{ people to party}$



$N = 2$ $((1) (2)) , ((1, 2))$

$N = 3$ $((1) (2) (3)) , ((1, 2), (3)) ,$
 $((1, 3) (2)) , ((2, 3) (1))$



$$way(4) = \frac{(4-1)}{2} \times \underline{\underline{way(2)}} = 6$$

RR

$way(n) = way(n-1) + (n-1) way(n-2)$

$[0-N]$

```
int dp[N+1] = {-1};
```

```
int way(int n) {  
    if (n <= 2) return n;
```

```
    if (dp[n] != -1) {  
        return dp[n];
```

```
    }  
    ans = way(n-1) + (n-1) * way(n-2);
```

```
    dp[n] = ans;
```

```
    return ans;
```

}

#US $\rightarrow N$

TRPS $\rightarrow O(1)$

$\boxed{TC = O(N)}$

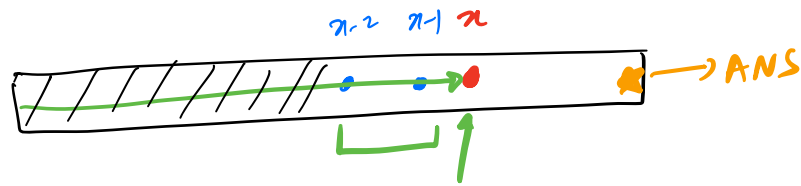
$\boxed{SC = O(N)}$

DL

$N=1 \rightarrow 1$

$N=2 \rightarrow 2$

Bottom Up A.



```
int dp[N+1];  
dp[1] = 1, dp[2] = 2;  
for (i = 3; i <= N; i++) {  
    dp[i] = dp[i-1] + (i-1) * dp[i-2];  
}  
return dp[N];
```

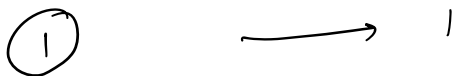
$TC = O(N)$
 $SC = O(N)$

Use 2 variables like fibonacci

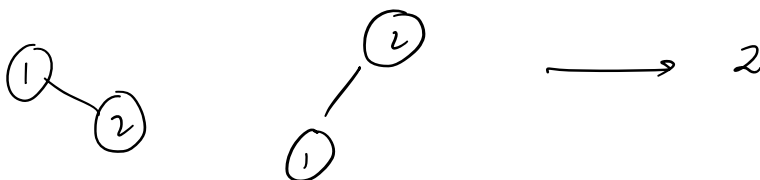
$SC \rightarrow O(1)$

Q Given N DISTINCT KEYS.
Find the # of BSTs you can form using these!

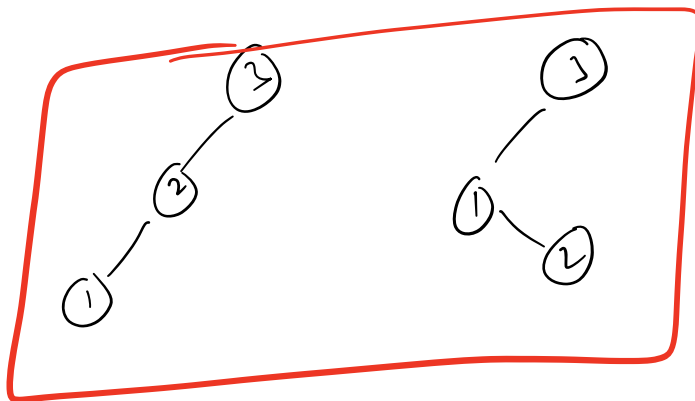
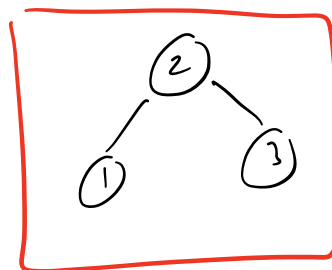
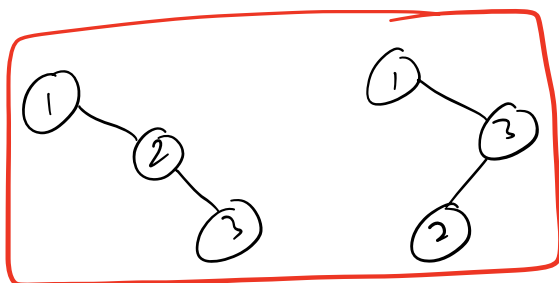
$N=1$



$N=2$

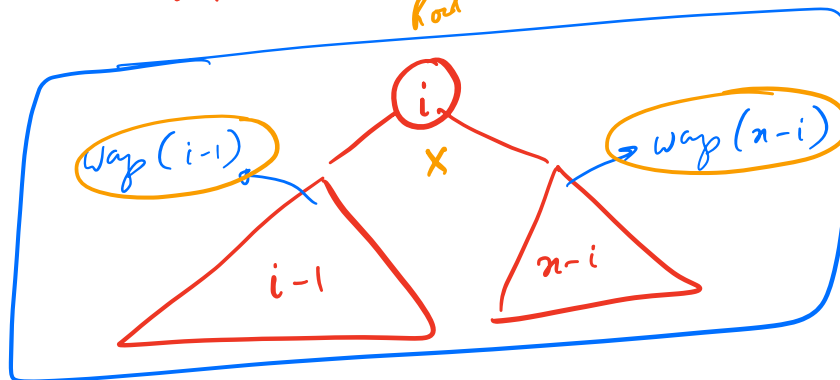
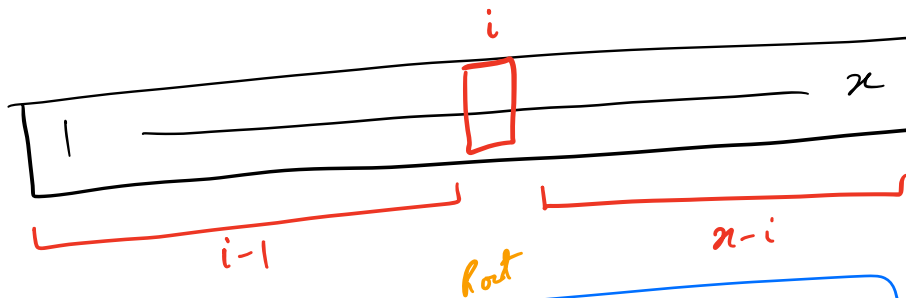


$N=3$

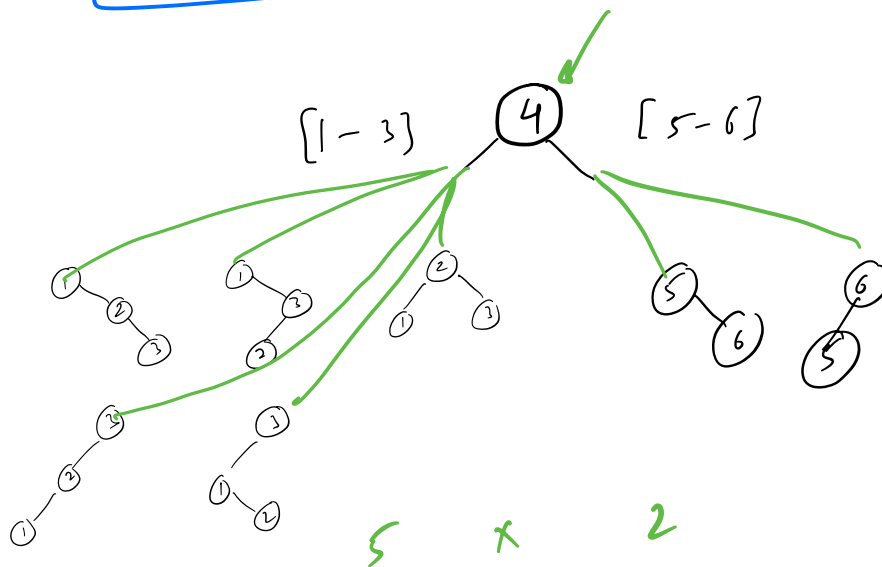


→ 5

way(n) → # of ways of forming BSTs with n keys



$n = 6$



$= 10$

$Way(n) : i \text{ is root} = Way(i-1) \times Way(n-i)$

RR

$$\text{ways}(n) = \sum_{i=1}^{i=n} \text{ways}(i-1) \times \text{ways}(n-i)$$

ans

O-N

```
int dp[N+1] = {-1};
int ways( int n) {
    if (n <= 1) ret 1;
```

```
    if ( dp[n] != -1)
        ret dp[n];
```

```
    ANS = 0;
    for ( i=1; i <= n; i++) {
        ANS += ways(i-1) * ways(n-i);
```

```
    }
    dp[n] = ANS;
    ret ANS;
```

```
>
```

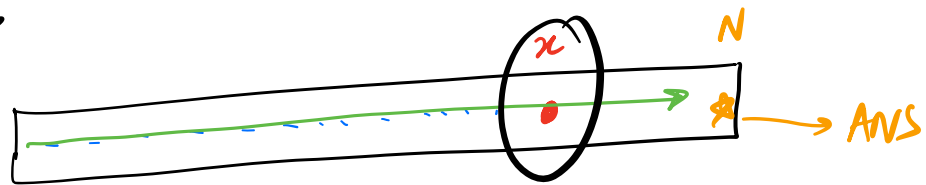
#US $\rightarrow N$

TRPS $\rightarrow O(N)$

TC = $O(N^2)$

SC = $O(N)$

Bottom up →



```
int dp[N+1];
```

```
dp[0] = 1;
```

```
f(x=1; x<=N; x++) {
```

```
    ans = 0;
```

```
    f(i=1; i<=x; i++) {
```

```
        ans += dp[i-1] * dp[x-i];
```

```
    }
```

```
    dp[x] = ans;
```

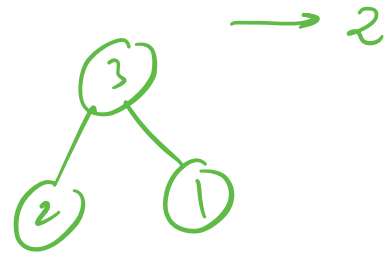
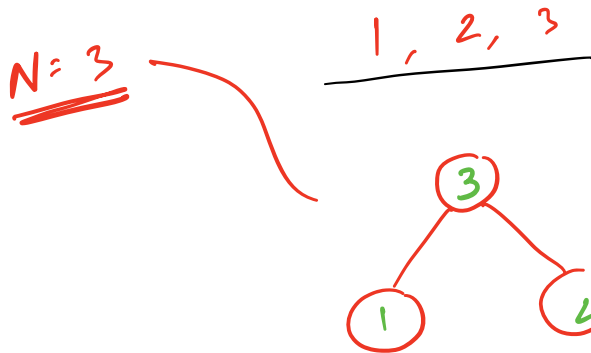
```
}
```

```
return dp[N];
```

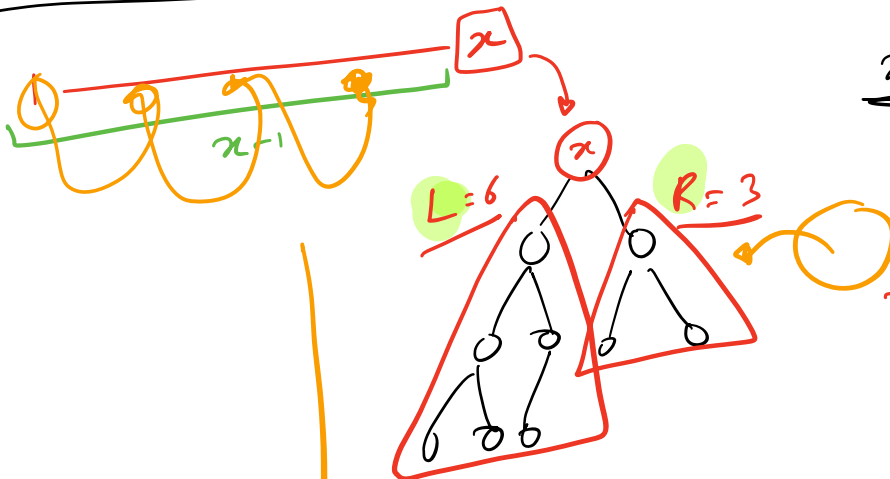
~~TC = $O(N^2)$~~

~~SC = $O(N)$~~

Q Given N distinct No's.
find the # of ways of forming MAX heap!



way(n) = # of ways of forming max heaps with n values



$n=10$

$$L+R = n-1$$

$$R = n-1-L$$

$${}^nC_r = {}^nC_{n-r}$$

way(n) = ${}^{n-1}C_L \times \text{way}(L) \times \text{way}(R)$

${}^{n-1}C_L$

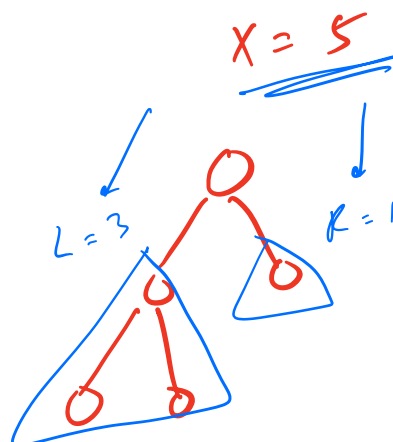
${}^{n-1}C_{n-1-L}$

${}^{n-1}C_R$

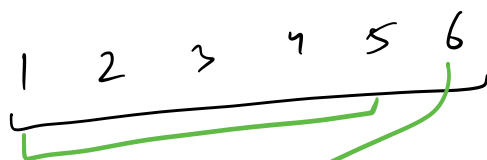
~~RR~~

$$\text{way}(n) = {}^{n-1}C_R \times \text{way}(L) \times \text{way}(R) \quad \checkmark$$

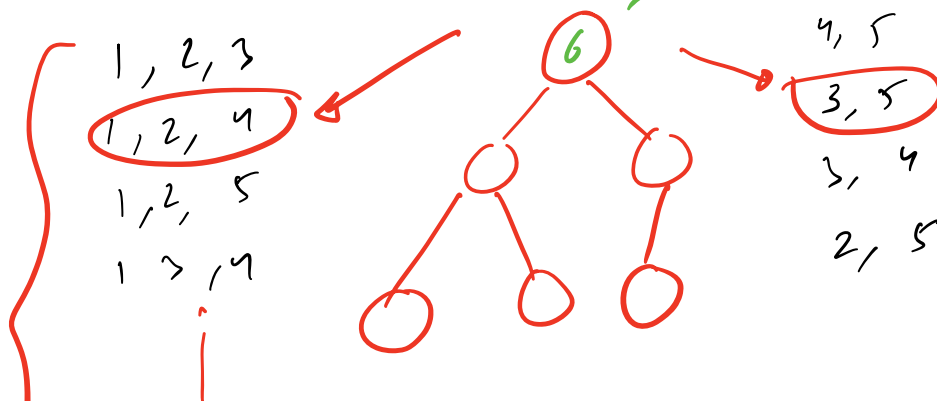
0-N



$X = 6$



$L = 3$
$R = 2$



$${}^{6-1}C_3 \times \text{way}(3) \times \text{way}(2)$$

HW
DP!

Q

Fractional Knapsack

You have a bag with capacity C
— N items

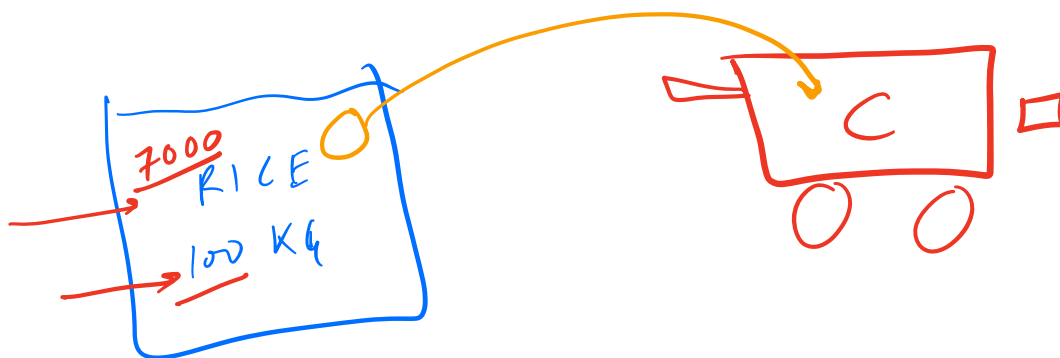
└→ 1. price $p[i]$
└→ 2. weight $w[i]$



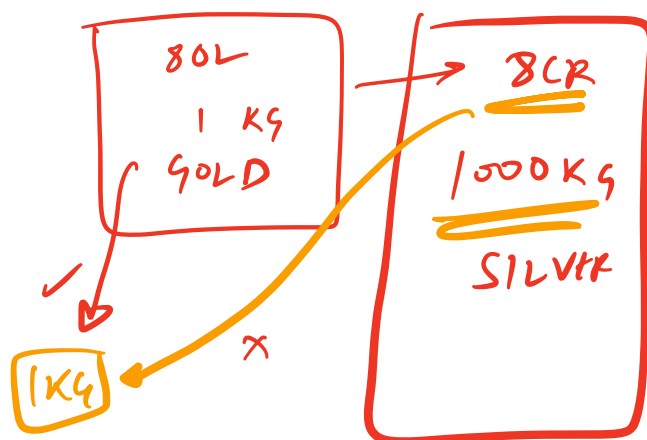
Find out the largest price you can have of the
the items collected in the bag!

→ You can pick fraction of items!

DAL
RICE
WHEAT
oil
i



Greedy!



O/I K NAPSACK

You have a bag with capacity
 — N items



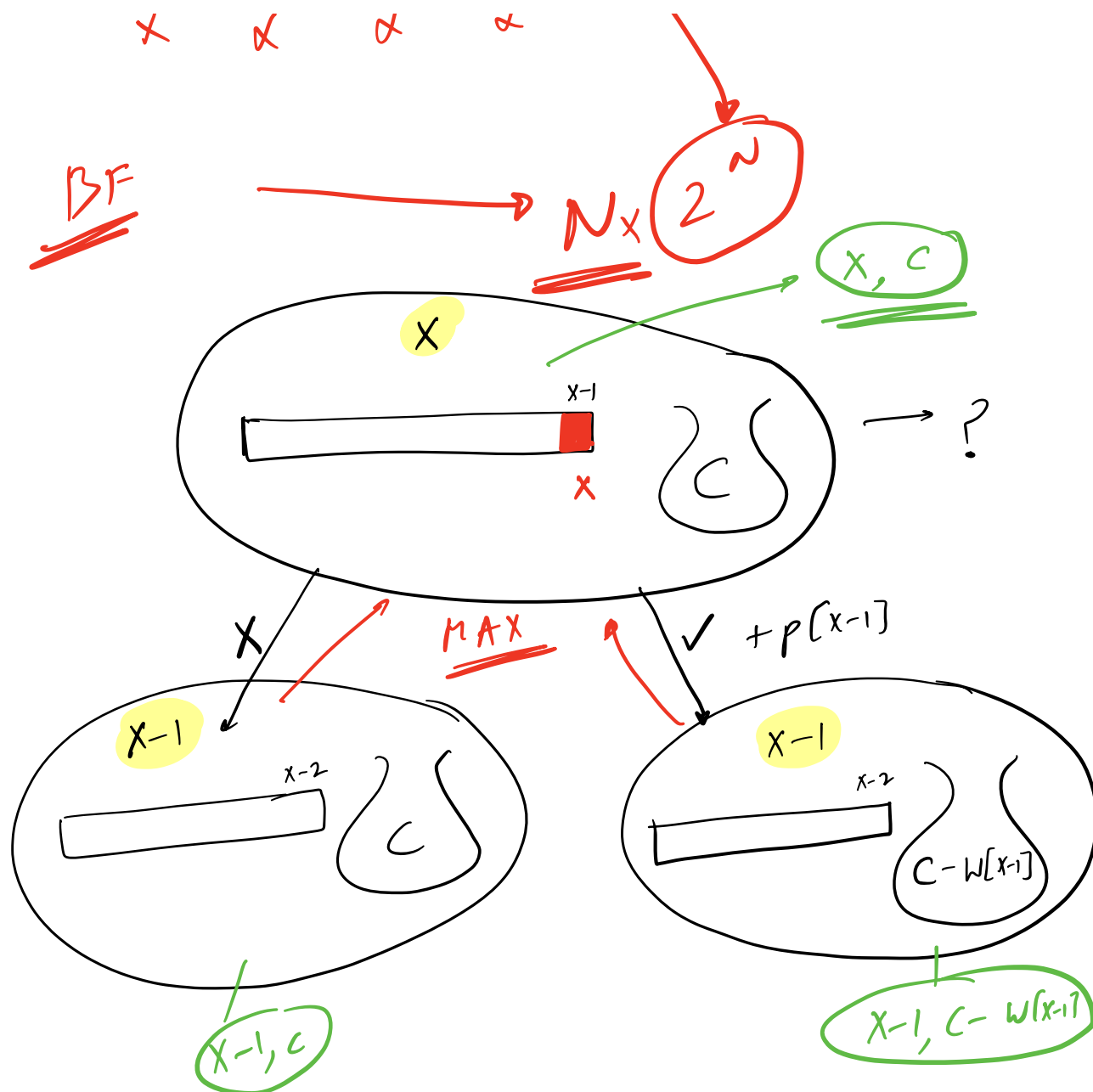
The diagram shows a large rectangle labeled 'X' in orange. Inside it is a smaller rectangle labeled 'C' in red. A green arrow points from the 'X' box to the 'C' box, indicating a relationship. Below the 'X' box, there are two smaller boxes, each containing a green arrow pointing down. A red arrow points from one of these boxes to the 'C' box. Another red arrow points from the bottom of the 'C' box to the bottom of the 'X' box. A green checkmark is placed near the bottom of the 'X' box, and a green 'X' is placed near the top of the 'X' box.

I) Generate all subsets

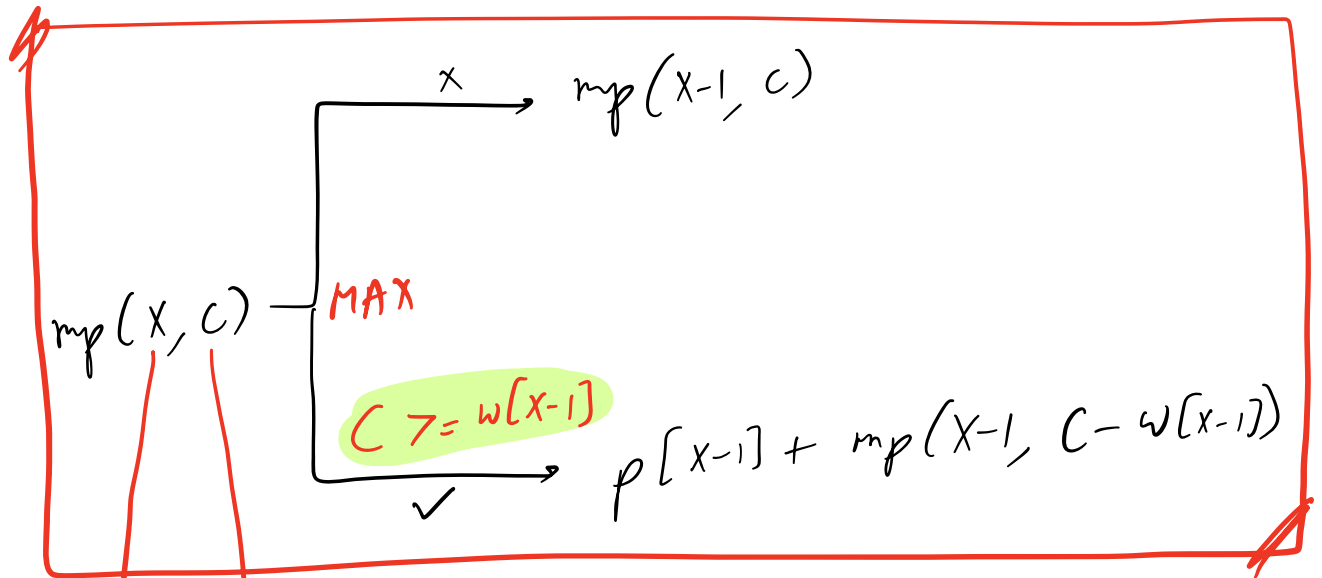
→ find the valid ones & take MAX.

MAX of total price!

1	2	3	4	5	6
✓	✓	✓	✓	✓	✓



$mp(x, c) \rightarrow$ the max price you can get
 considering the items $[0 - x-1]$
 & rem. cap of the bag $\Rightarrow c$



$1-N$ $0-C$

x

$N \times C$ UNIQUE STATES

```
int dp[N+1][C+1] = {-1};
```

```
int mp(x, c) {
    if (x == 0) return 0;
```

```
    if (dp[x][c] != -1) {
        return dp[x][c];
```

```
    }
    ans = mp(x-1, c);
```

```
    if (c >= w[x-1]) {
```

```
        ans = max(ans, mp(x-1, c - w[x-1]) + p[x-1]);
```

```
    }
```

$dp[x][c] = \text{ANS};$

$\text{ret ANS};$

}

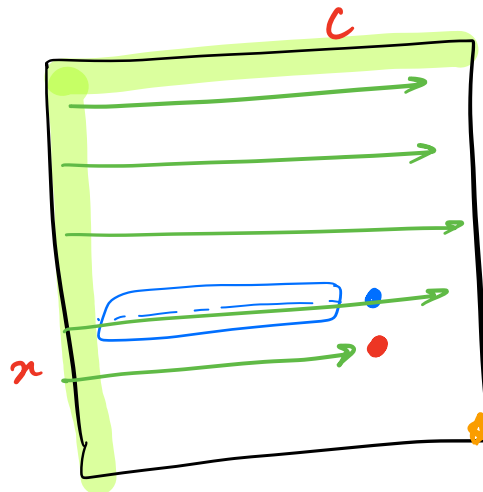
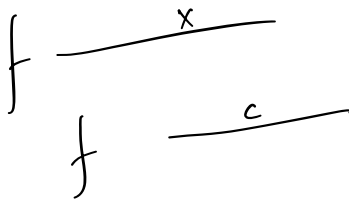
#US \rightarrow NC

TRPS $\rightarrow O(1)$

$TC = O(NC)$

$SC = O(NC)$

④ Bottom Up

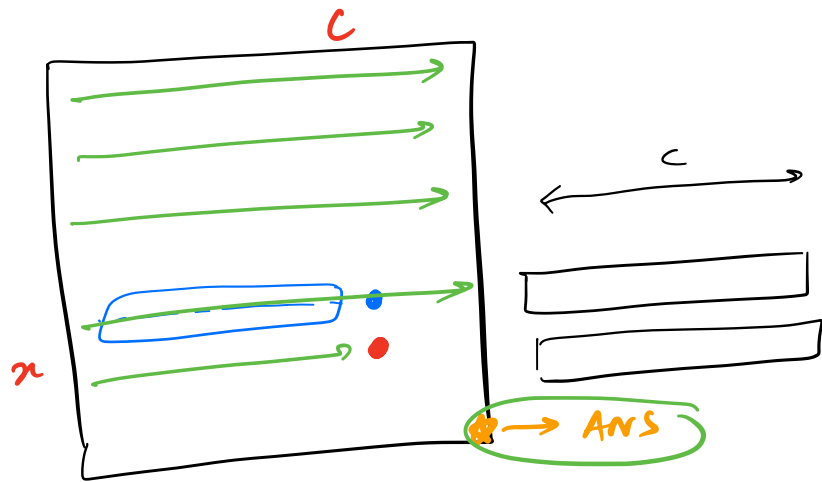


Ans

$dp[N][C]$

$TC = O(NC)$

$SC = O(NC)$



$$Sc = O(c)$$