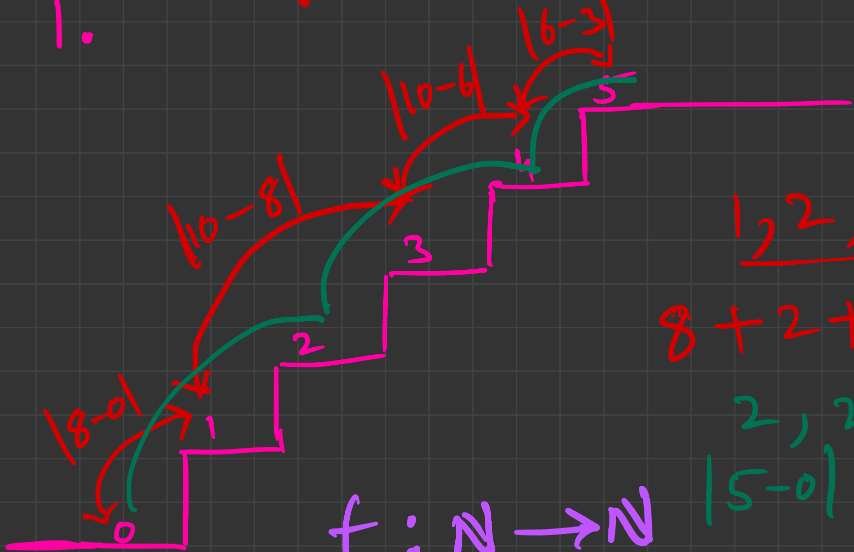
08/08/22

0	1	2	3	4	5
0	8	5	10	6	3

Home Work

1.



1, 2, 1, 1

$8+2+4+3=\underline{17}$

2, 2, 1

$|5-0|+|6-5|+|6-3|$

$f : \mathbb{N} \to \mathbb{N}$

$f \mapsto$ Total no. of ways

1 way : 2, 1, 2

$5+1+3=\underline{9}$

2 way : 1, 1, 2, 1

Least

3 way : 2, 2, 1

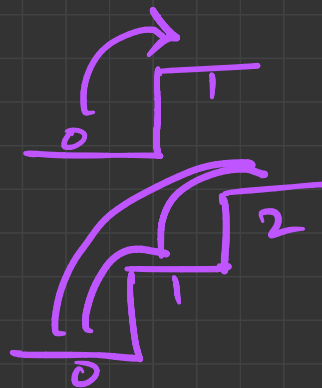
4 way : 1, 1, 1, 1, 1

⋮

$f(n)$ = "Total No. of ways to climb "n" stairs subject to climbing 1 or 2 stairs at a time"

$n=1, \ f(1)=1$

$n=2, \ f(2)=2$



$$\boxed{f(n) = ?}$$

2.

3	1		11	10
6	9	0	3	
1	2		8	6
10	-	14	0	15

✓

R ✓ →

↓ D ✓

25 X 30

4 x 5

RRDRRDD

DRDRRDR

RRRRDDD ✓

{D, D, D, D,
R, R, R}

max = ?

$$\frac{7!}{3! \times 4!}$$ X

3. "set"

$\{1, 2, 5, 9, 7\}$ ✓

✓ $S = \{1, 2, 5\}$ sum = 8

✓ $S = \{7, 1\}$

- can we get a subset?

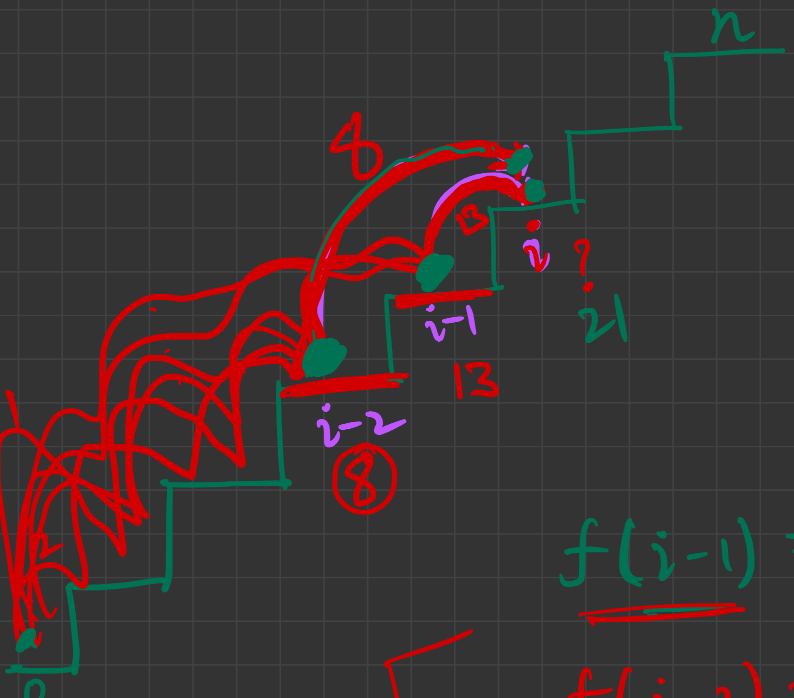
Yes | No

- How many such subsets?
- How many elements in

largest subset?

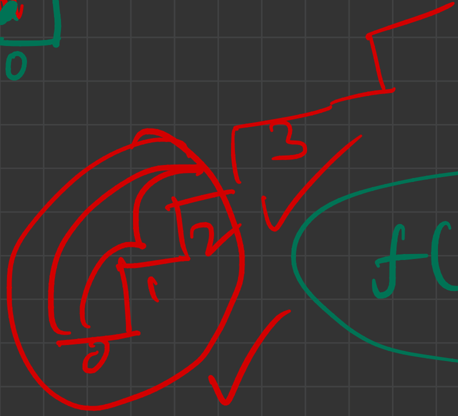
- How many elements in second largest subsets?

$f(n) = ?$

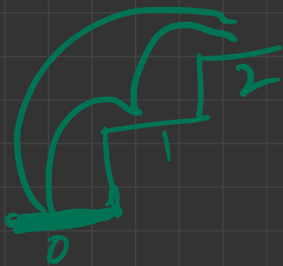


$f(i-1) = 13$

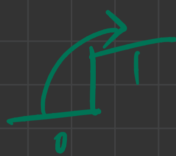
$f(i-2) = 8$



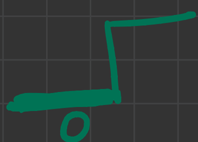
$$f(i) = f(i-1) + f(i-2)$$



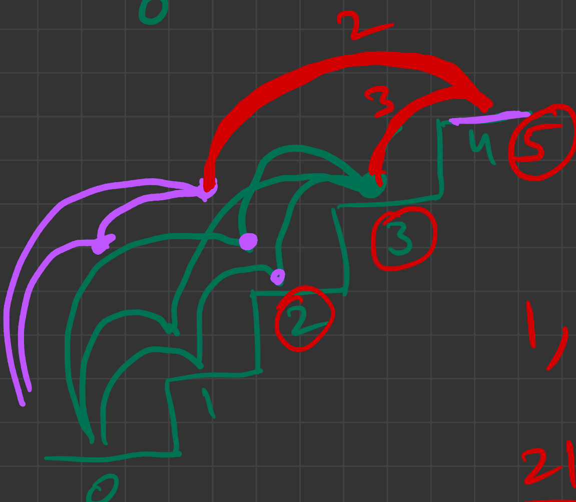
$$f(2) = f(1) + f(0)$$
$$= 1 + 1$$
$$= 2$$



$$f(3) = f(2) + f(1)$$
$$= 2 + 1$$
$$= 3$$



$$f(4) = f(3) + f(2)$$
$$= 3 + 2$$
$$= 5$$



6

1, 1, 2, 3, 5, 8, 13,

21, 34, ----

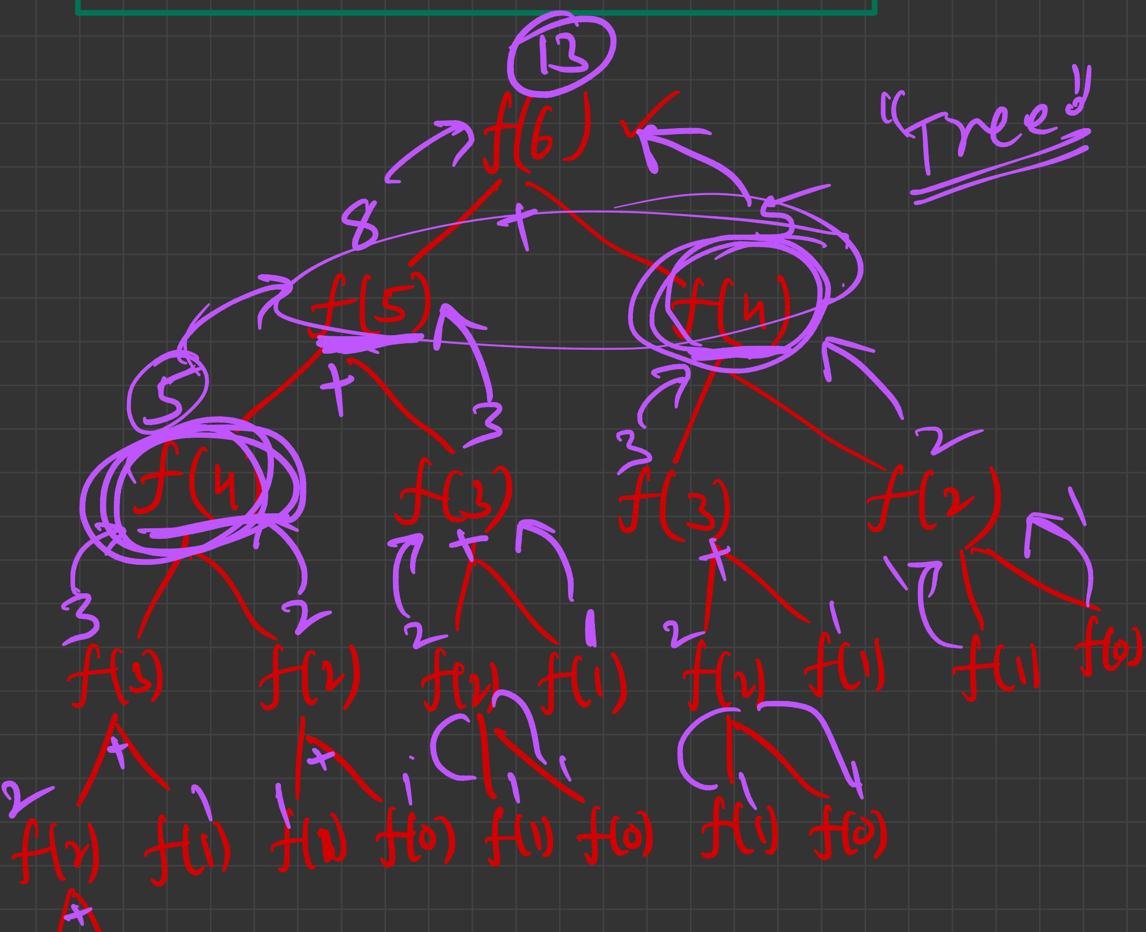
$$\boxed{f(n) = f(n-1) + f(n-2)}$$

pseudo ≈ C++

```
int f(int n){
    if(n==0 || n==1)
        return 1;
    return f(n-1)+f(n-2);
```

$f(0)=1$
$f(1)=1$

"Tree"



f(1) f(0)

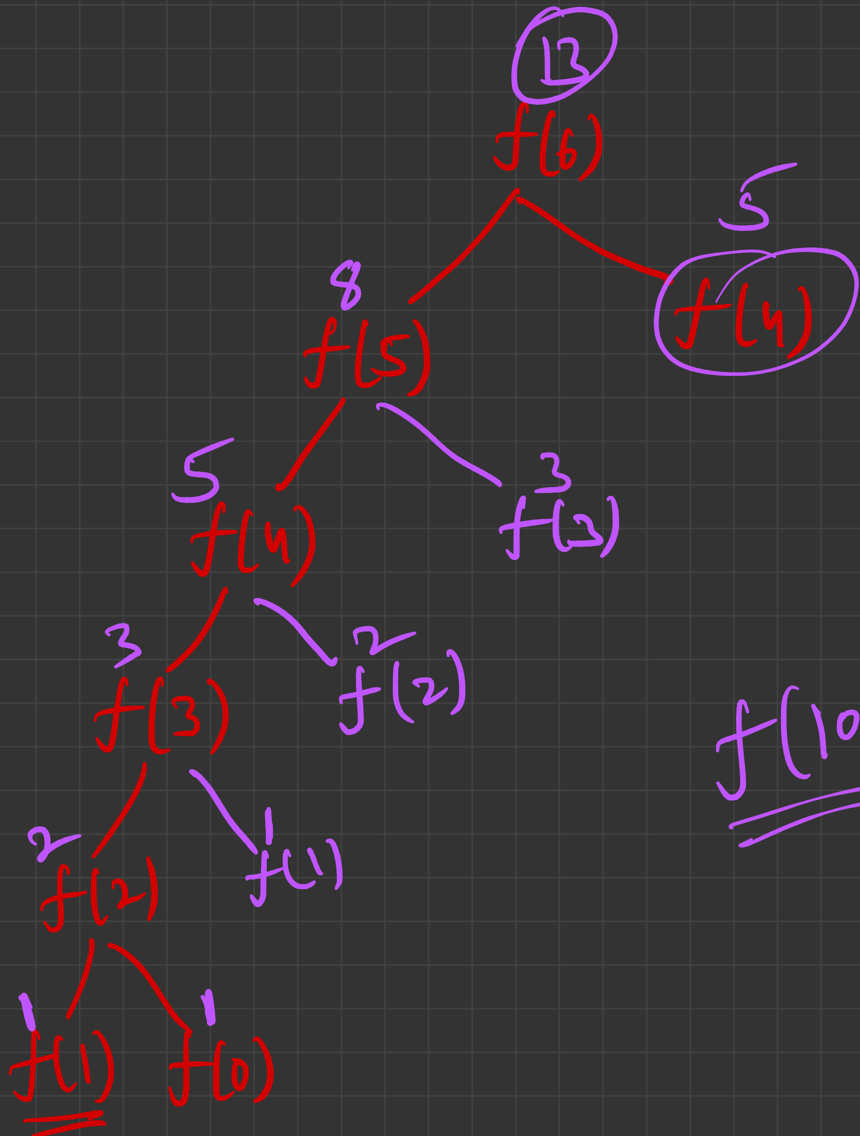
"stack"

"call stack"

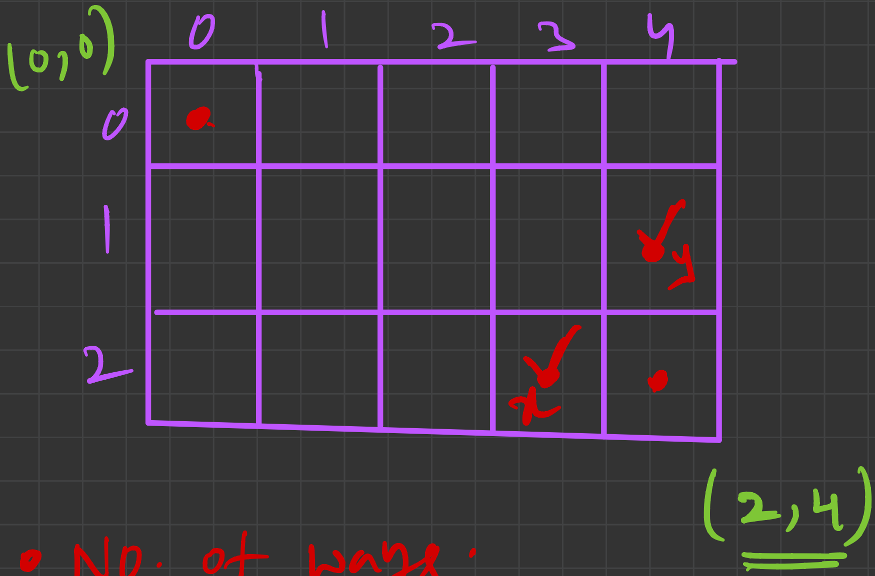
```
f(n, A) {
    if (n==0 || n==1)
        return 1;
    if (A[n] != -1)
        return A[n];
    return A[n] = f(n-1) + f(n-2);
}
```

A = [-1, -1, -1, -1]

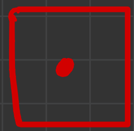
"memoization"



$f(100)$

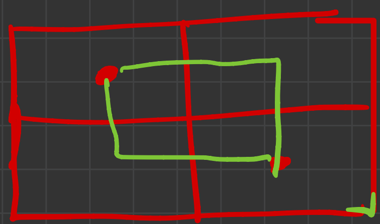


• No. of ways:



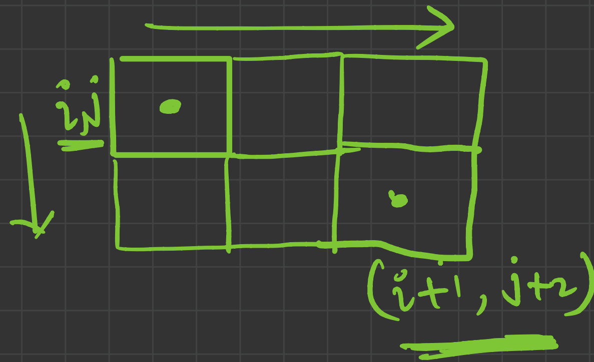
1 way

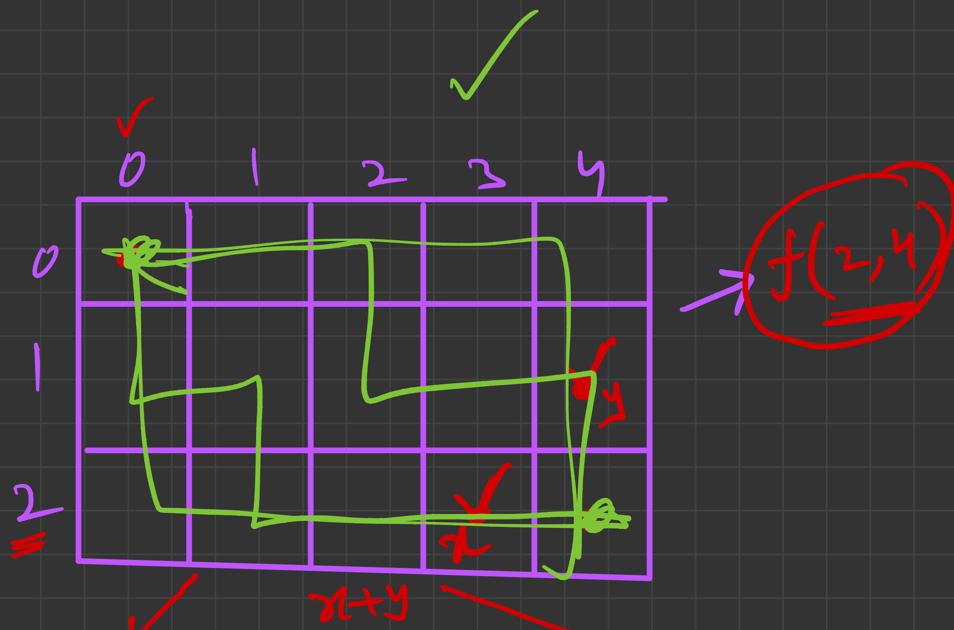
1 way

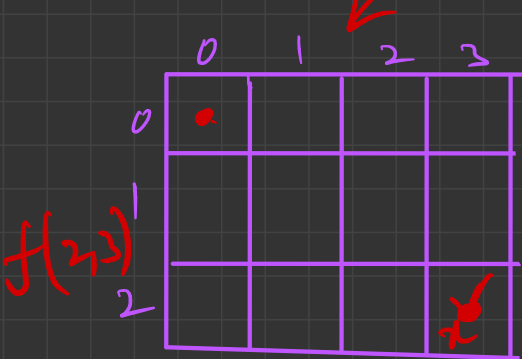


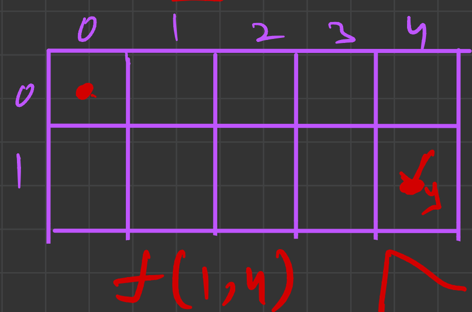
2 ways

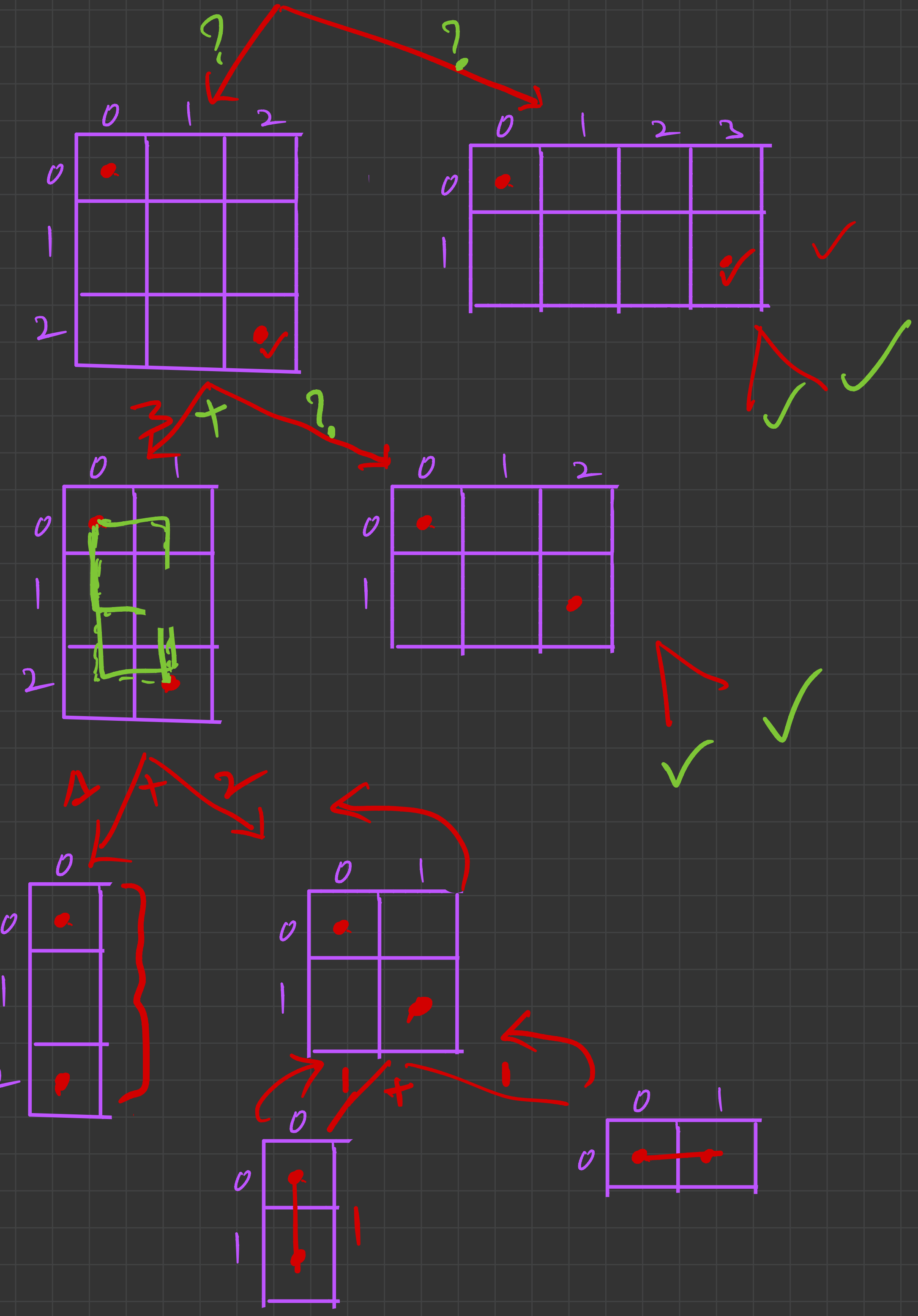
$f(i, j)$

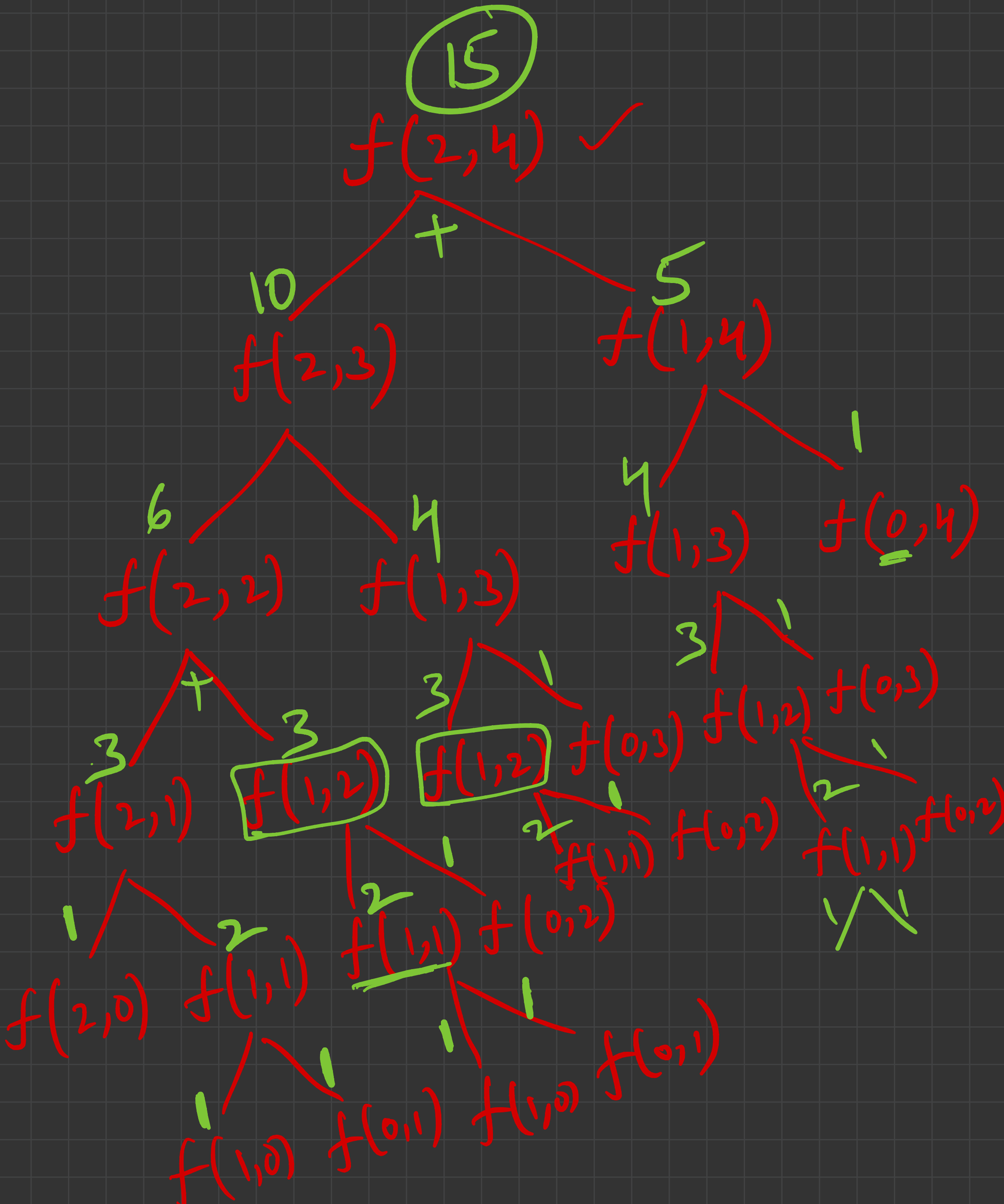












15
f(2,4)
10
f(2,3)
5
f(1,4)
6
f(2,2)
4
f(1,3)
f(1,3)
f(0,4)
3
f(2,1)
f(1,2)
f(1,2)
f(0,3)
f(1,2)
f(0,3)
f(2,0)
f(1,1)
f(1,1)
f(0,2)
f(1,1)
f(0,2)
f(1,1)
f(0,2)
f(1,0)
f(0,1)
f(1,0)
f(0,1)

$$f(i,j) = f(i-1,j) + f(i,j-1)$$

```
int f(int i, int j){
    if(i==0 || j==0)
        return 1;
    return f(i-1,j)+f(i,j-1)
}
```

✓

	0	1	2	3
0	-6	5	0	-10
1	20	-5	[illegible]	-8
2	3	[illegible]	70	9
3	5	6	1	18

m_1

m_2

+ve

m_1, m_2

$\boxed{f(3,3)}$ ✓

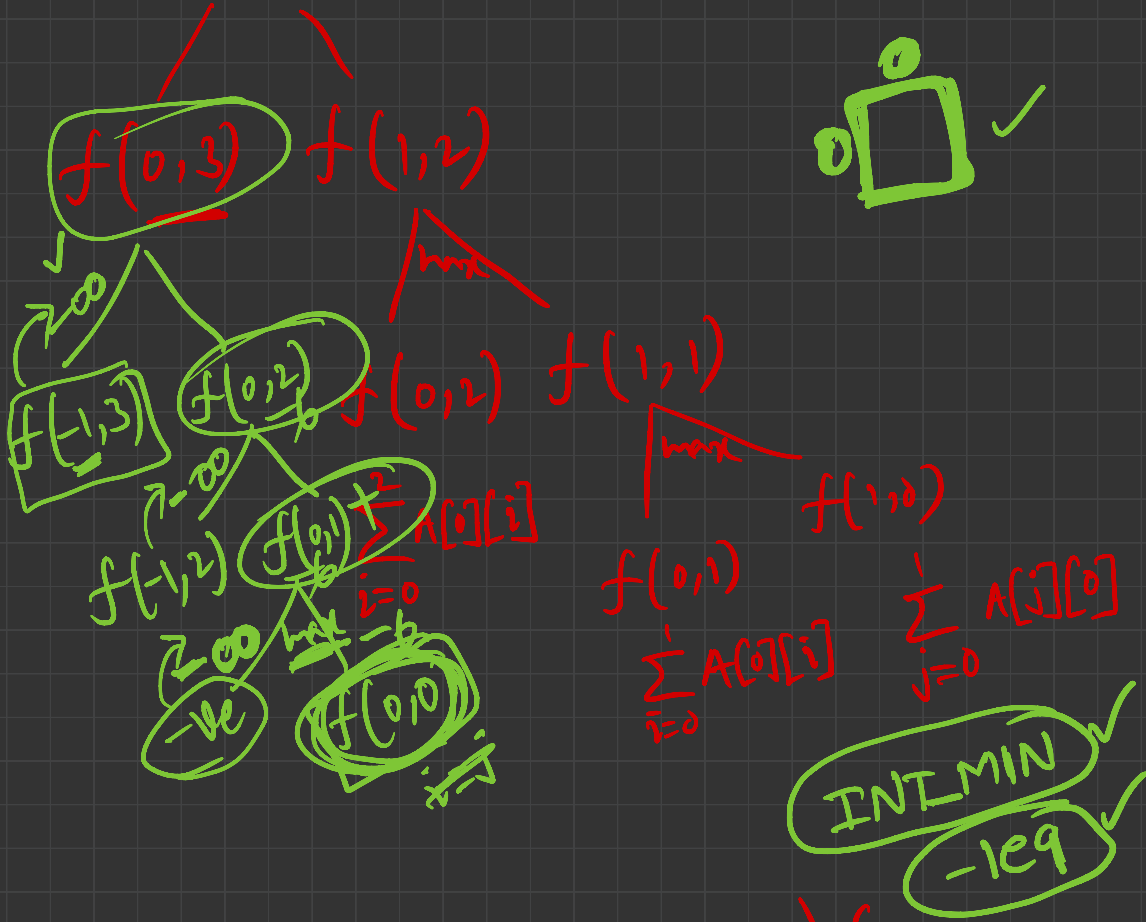
max

$f(2,3)$ $f(3,2)$

max $+ A[2][3]$ max $+$

$f(1,3)$ $f(2,2)$ $f(\ )$ $f(\ )$

max




```
int f(int i, int j){
    if(i==0){
        sum=0;
        for(k=0; k<=j; k++)
            sum += A[0][k];
        return sum;
    }
```

```
if (j==0){
    sum=0;
    for (k=0; k<=i; k++)
        sum += A[k][0];
    return sum;
}

return max(f(i-1,j), f(i,j-1))
                        + A[i][j]

int f(int i, int j){
    if( i==0 && j==0){
        return A[0][0];
    if( i<0 || j<0 }
        return INT_MIN;
```

return $\max(f(i-1,j), f(i,j-1)) + A[i][j]$

✓ $A = \{1, 5, 3, 4\}$ ✓

Distinct

List

Array

✓ $A = \{1, 1, 5, 3, 4, 4\}$ "finite"

✓ $A = \{\underset{\infty}{1}, \underset{\infty}{5}, \underset{\infty}{3}, \underset{\infty}{4}\}$ ✓

- $A = \{\underset{\infty}{2}, \underset{\infty}{5}, \underset{\infty}{3}\}$, $\underline{\underline{T = 11}}$

- Is it possible? ✓

T/F

- How many different solutions?

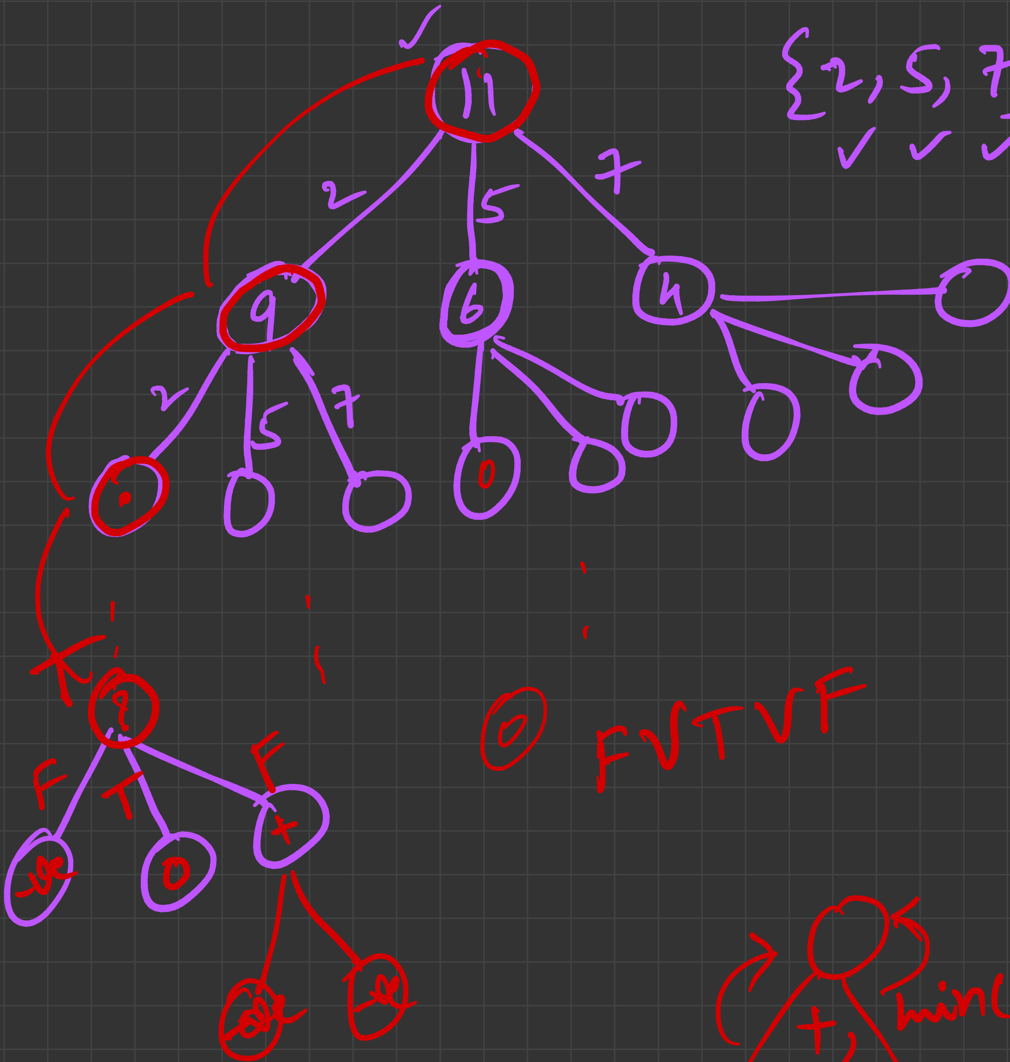
$$f(x) = \begin{cases} 1 & x = 0 \\ 0 & x < 0 \\ \vee f(x-a) & \text{o.w} \checkmark \end{cases}$$

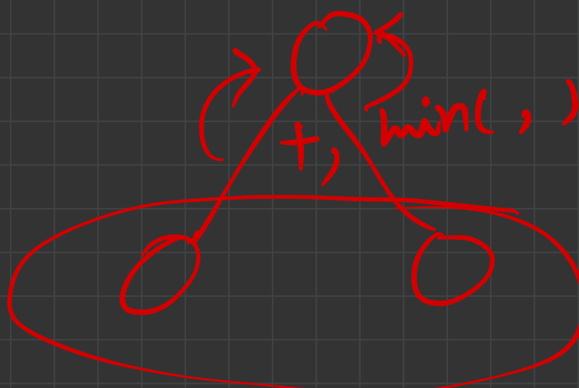
$$f(11) = f(11-2) \vee f(11-5) \vee f(11-7)$$

$$= f(9) \vee f(6) \vee f(4)$$

$\vdots$

$\{2, 5, 7\}$





$+$: Int

min(,) : Intersn

$\vee$: Boolean

```
f(T){
  if (T == 0)
     return True;
  if (T < 0)
     return False;
  result = False
  for (auto a : A){
     result = result ∨ f(T-a);
  return result;
}
```

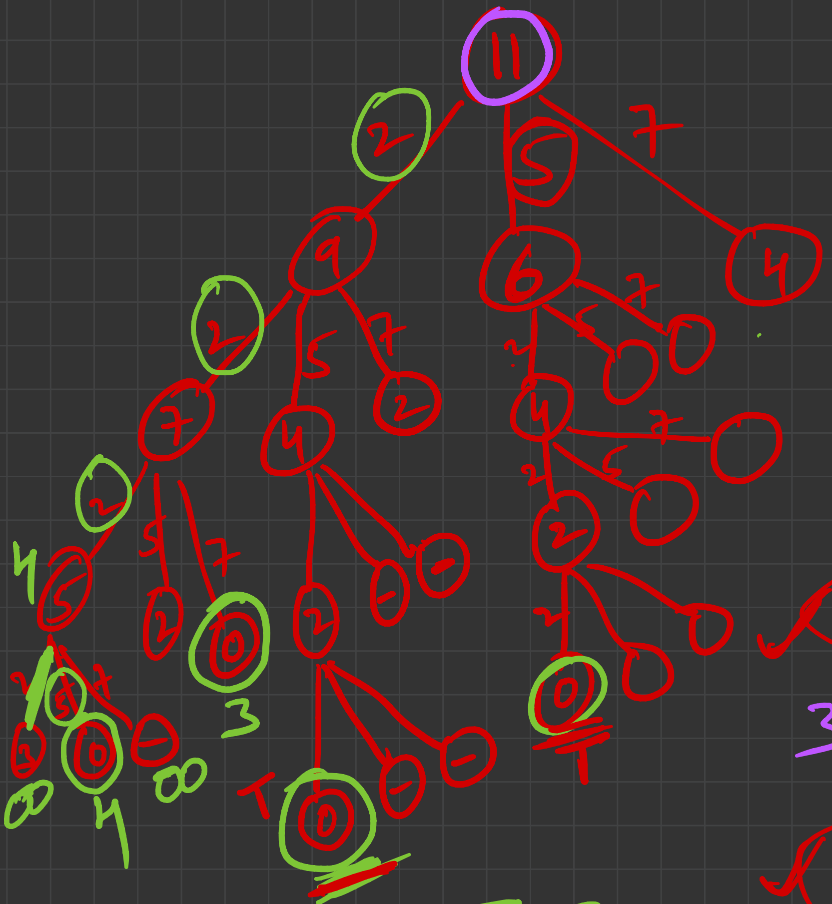
$$\bigvee_{a \in A} f(T-a)$$

```
result = False
for( ________ )
  result = result ∨ f( )
```

$$\sum_{n=1}^{5} n$$

```
sum = 0
for( ________ )
  sum += n
```

$\{2, 5, 7\}$



4

$2+2+2+5$

3 $2+2+7$

$5+2+2+2$

$T \vee F = T$

$1+0+1=2$

T, F

1, 0

Dupnic

min

0	-ve	0	0
=	=	=	=
1	0	1	1

$1 + 0 + 1 + 1$

$$f(T) = \begin{cases} 1 & T = 0 \\ 0 & T < 0 \\ \sum\limits_{a \in A} f(T-a) & \text{otherwise.} \end{cases}$$

```
f(T) {
    if (T == 0) return 1
    if (T < 0)  return 0
    ans = 0;
    for (auto a : A)
        ans += f(T-a)
    return ans
}
```

• "minimum number of entries"

$$f(T) = \begin{cases} 0 & T = 0 \\ \infty & T < 0 \\ \min\limits_{a \in A} \{ f(T-a) + 1 \} \end{cases}$$

$\{2, 5, 7\}$

$$f(9) = \min\left(f(7)+1,\ f(4)+1,\ f(2)+1 \right)$$

$$= \min\Big(\min\left(f(5)+1,\ f(2)+1,\ 0+1\right),$$

$$\min\left(f(2)+1,\ \infty+1,\ \infty+1\right),$$

$$\min(0+1,\ \infty+1,\ \infty+1))$$

$$= \min\Big(\min\big(f(5)+1,\ f(2)+1,\ 1\big),$$

$$\min\big(f(2)+1,\ \infty,\ \infty\big),$$

$$1\Big)+1$$

$$= \min\Big(\min\big(\min(f(3)+1,\ 0+1,\ \infty)+1$$

$$\min(\min(0+1,\infty,\infty)+1)+1,$$

$$1\Big)+1$$

$$= \min\Big(\min\big(\min(f(1)+1,\ \infty,\infty)+1\big)+1\big)+1)$$

$$,\ 3,\ 2\Big)$$

$= \min\{\infty, 3, 2\}$

2

```
f(T){
   if(T==0) return 0
   if (T<0) return INT_MAX
     mini = INT_MAX
   for(auto a : A){
         mini = min(f(T-a)+1)
return mini
```

- Is subset sum possible?
- Num of ways
- least num of entries.

$$\{\overset{\infty}{2}, \overset{\infty}{5}, \overset{\infty}{7}\}$$

✓ • Is change possible for T?

$T = 13$ — True / False.

✓ • How many different ways to produce the coins for T?

✓ • Least num of coins for T?

$T < 0$

```
f(T) {
    if T == 0 :
        return True
    if T < 0 :
        return False
    Ans = False
    for (auto coin : A) :
        Ans = Ans V f(T-coin)
    return Ans
```

```
f(T) {
    if (T==0)
        return 1
    if (T<0)
        return 0

    sum = 0
    for(auto coin : A){
        sum = sum+f(T-coin)

    return sum
```

Duplicate

{1, 2, 3}

✓ 5 = 2+3

✓ 5 = 3+2

```
f(T) {
    if T==0:
        return 0
    if T<0:
        return ∞
    mini = ∞
    for(auto coin : A){
        mini = min(mini,
                   f(T-coin)+1)
    return mini
```

✓ • Draw Tree diagram

✓ • Look at Nature of subproblems

✓ • Decide base cases

✓ • Pseudo code

✓ • Implement

$$A = \{ \overset{\infty}{2}, \overset{\infty}{3}, \overset{\infty}{5} \}$$

$$\boxed{A = \{ \overset{\checkmark}{2}, \overset{\checkmark}{3}, \overset{\checkmark}{5} \}}$$ ✓

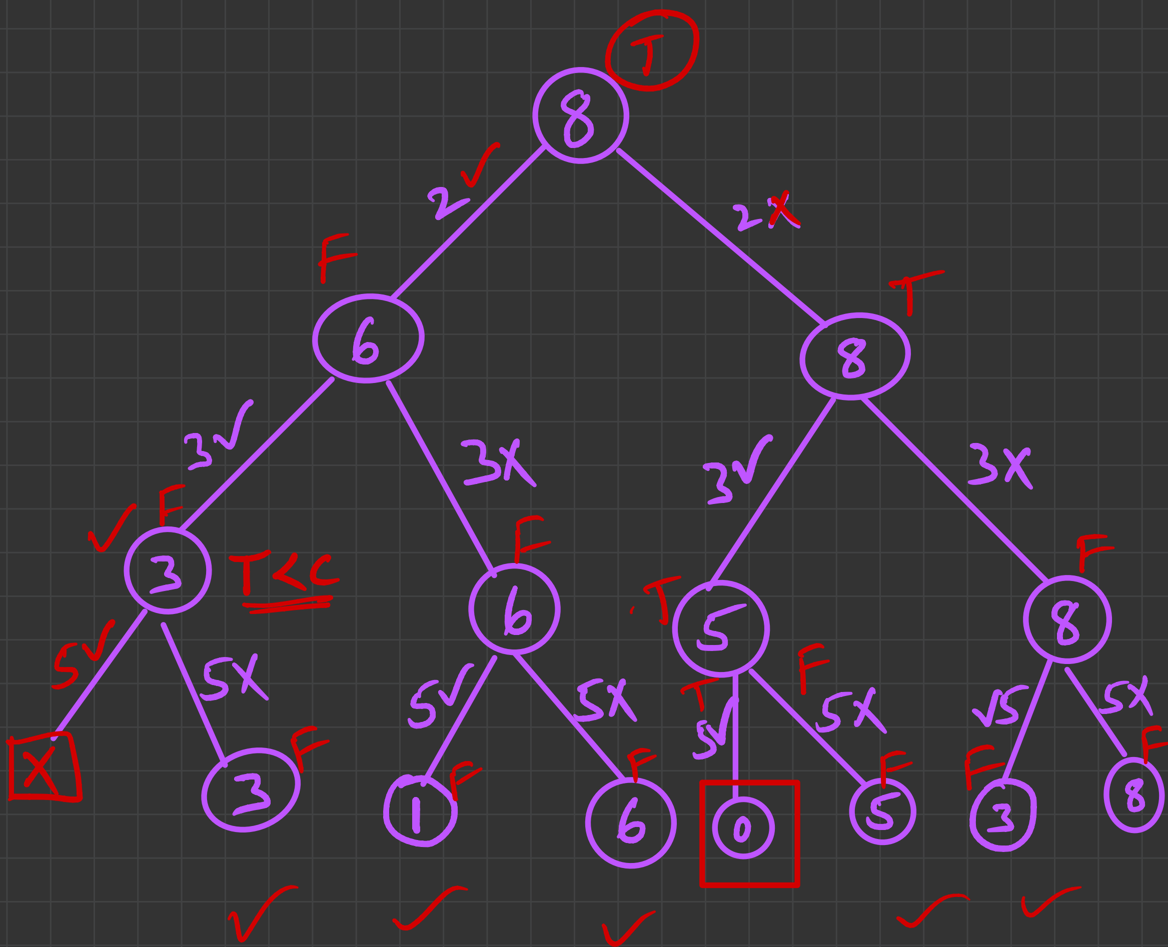
• Is subset sum Possible

$$T \checkmark \Rightarrow \underbrace{\{ \quad \}}_{S} \subseteq A$$

$$\boxed{\sum S = T}$$ ✓

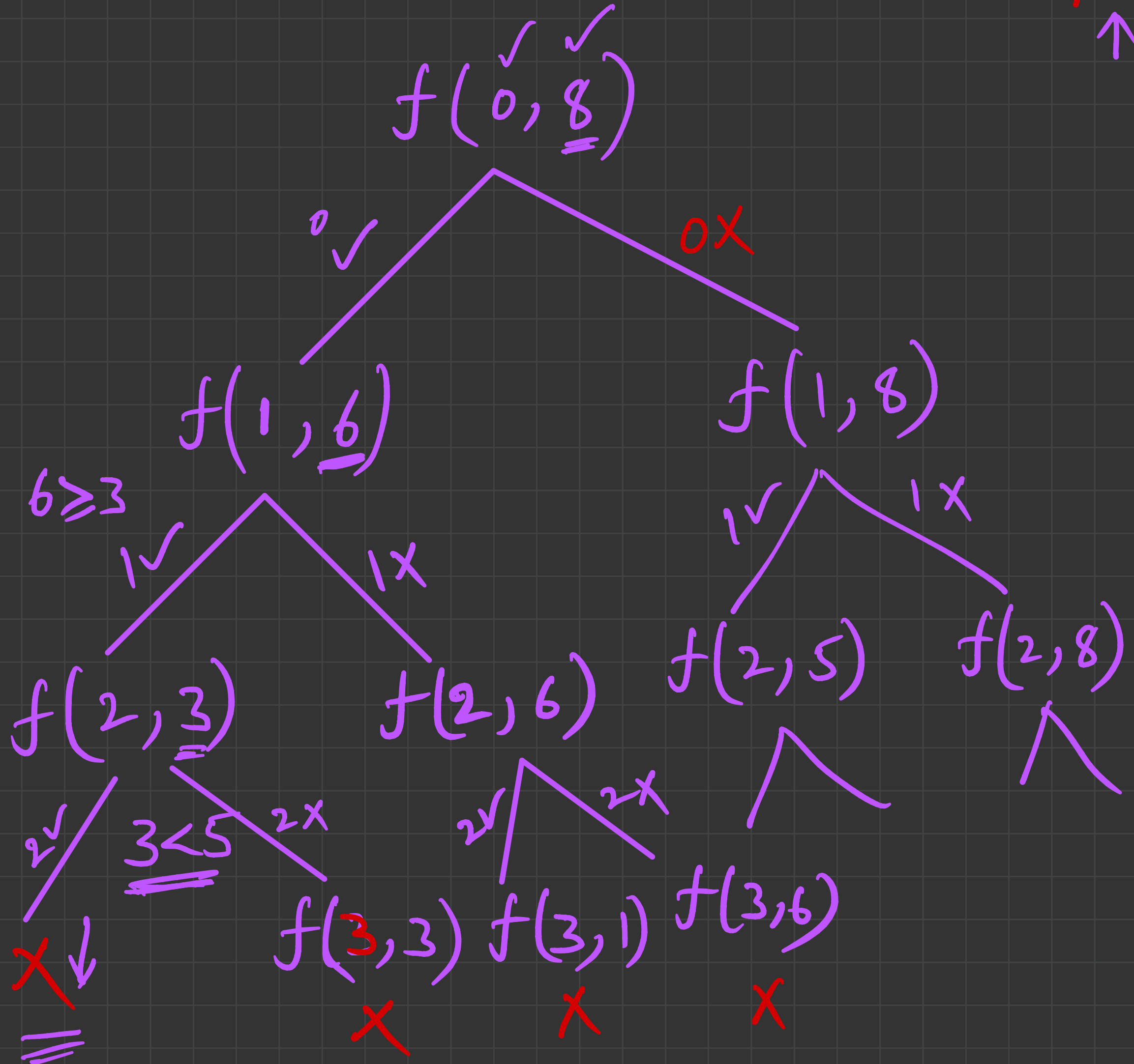
$A = \{2, 3, 5\}$

$T = 8$



$f(i, T, A)$

$$A = \{\overset{0}{2}, \overset{1}{3}, \boxed{\overset{2}{5}}\}$$

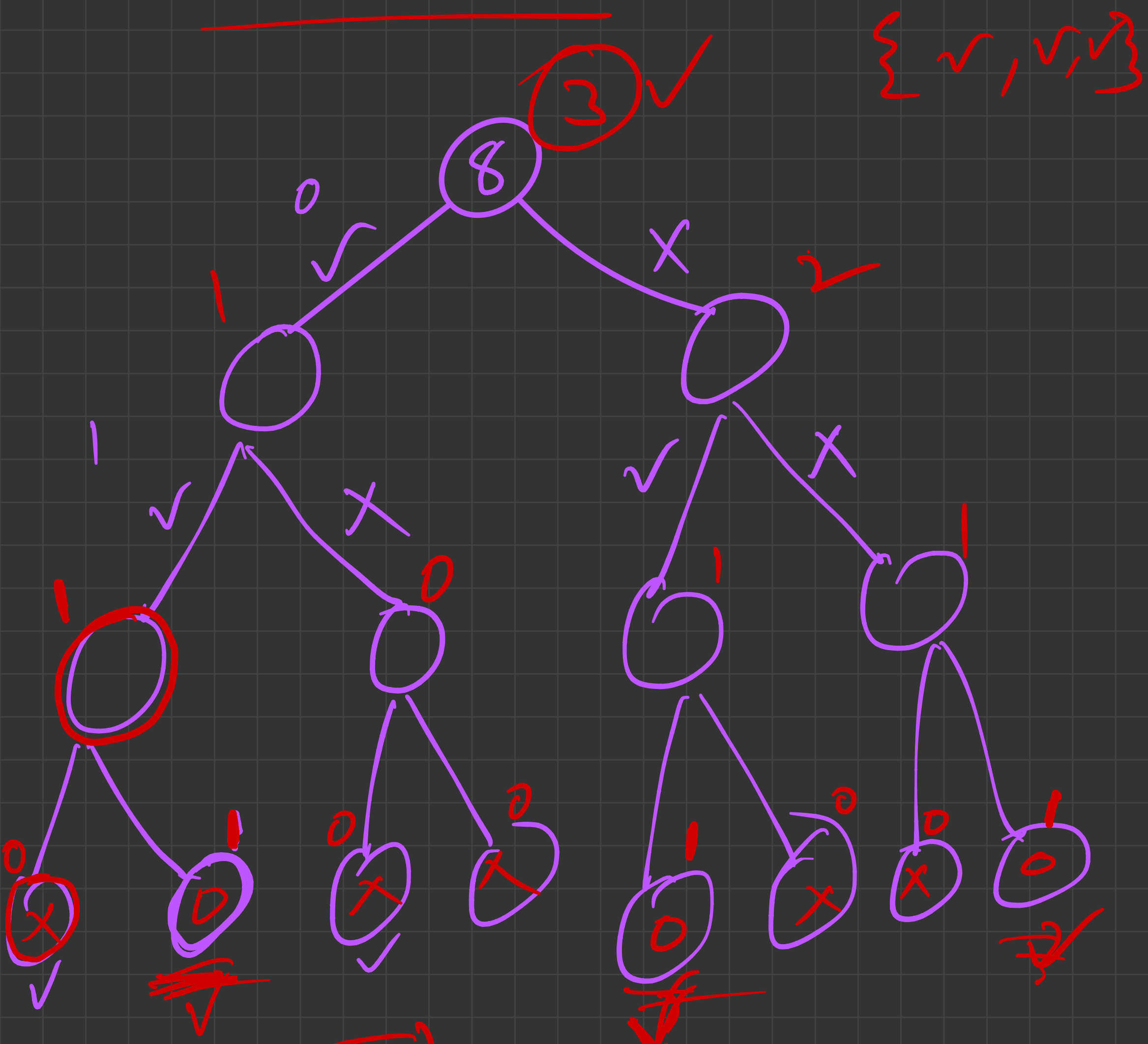


```
f(i, T, A) {
    if (T == 0) : return True
    if (i >= size(A)) return False.
    if (T < 0) : return False

    return f(i+1, T-A[i], A) V
           f(i+1, T, A)
```

```
f(i, T, A) {
    if (T == 0) : return True
    if (i >= size(A)) : return False.

   if (T ≥ A[i])
      Take = f(i+1, T-A[i], A)

   NotTake = f(i+1, T, A)

   return Take V NotTake
```

✓ • No of subsets to produce T ✓

✓ • <u>No of elements in smallest subset for sum T.</u>



$\boxed{\vee}$ $\boxed{+}$

```
f(i, T, A) {
  if(T==0) return 1
  if(i >= size(A)) return 0

  if(T >= A[i]) {
      Take = f(i+1, T-A[i], A)
  NotTake = f(i+1, T, A)
  return Take + NotTake
```

$f(i, T, A)\{$

$\text{if}(T \geq A[i])$

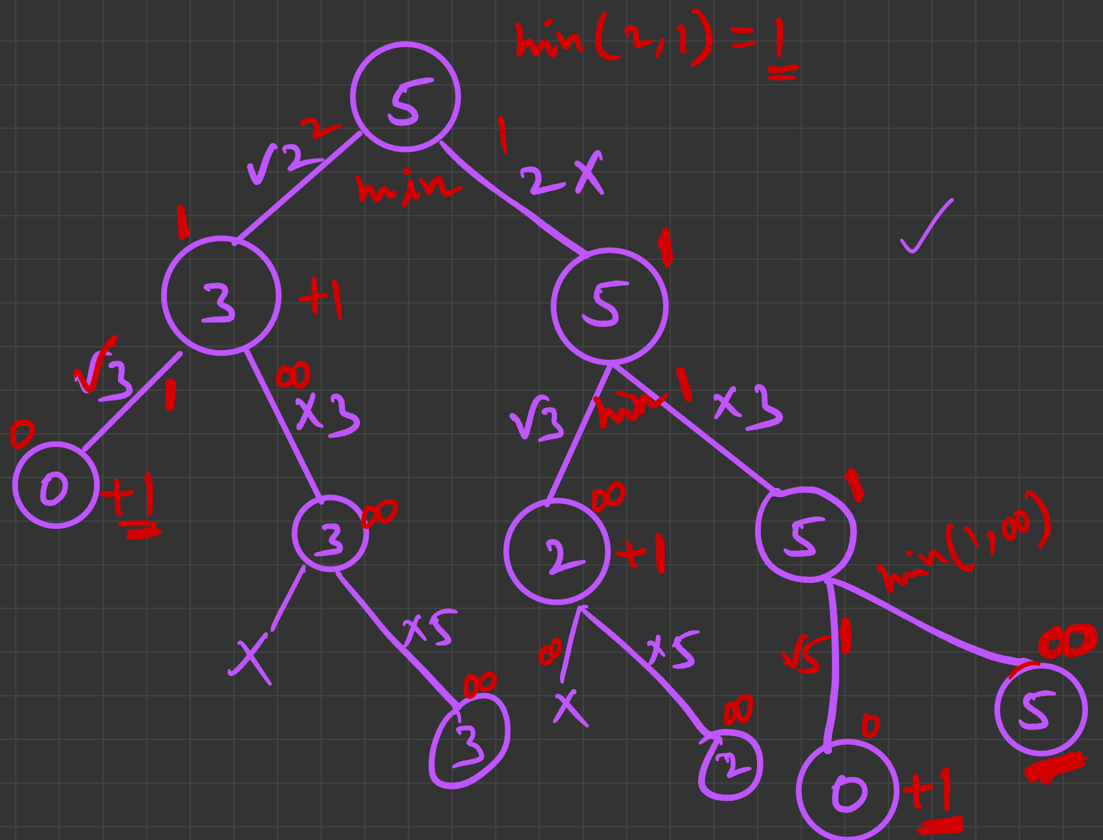
$\text{Take} = f(\underline{\underline{i+1}}, \underline{\underline{T-A[i]}}, A) + \underline{\underline{1}}$

$\text{NotTake} = f(\underline{\underline{i+1, T}}, A)$

$\underline{\underline{1D}}, \underline{\underline{2D}}$

$\{2, 3, 5\}$

$T = 5$



min(2,1) = 1
5
√2
2
min
1
2x
3
+1
5
1
√3
1
∞
x3
√3
min
x3
0
0
+1
3
∞
2
∞
+1
5
min(1,∞)
x
x5
∞
x
x5
√5
∞
3
∞
2
0
5
0
+1

```
f(i, T, A) {
    if T==0 : return 0
    if  i ≥ size(A) : return ∞

    if (T ≤ A[i])
        Take = f(i+1, T-A[i], A)
    NotTake = f(i+1, T, A)
    return min(Take, NotTake)
```

A, T

- no. of elements in largest subset.
- Is it possible to produce a subset with K-no. of elements
- no. of elements in second smallest "

① $\{\overset{\infty}{2}, \overset{\infty}{3}, \overset{\infty}{5}\}$ ✓

② $\{\overset{1}{2}, \overset{1}{3}, \overset{1}{5}\}$ ✓

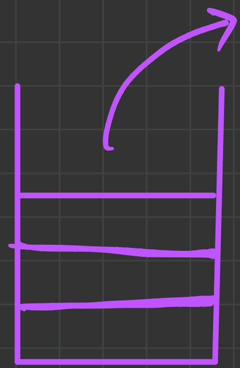
③ $\{2, 2, 3, 5, 5, 5\}$??

$\vee$ — Boolean ✓ ✓

$+$ — No. sol^n ✓ ✓

min — minimum sol set ✓ ✓

"STACK:"



"LIFO"

$\{2, 3, 5\} \longrightarrow T=5$

"solution" "subset"

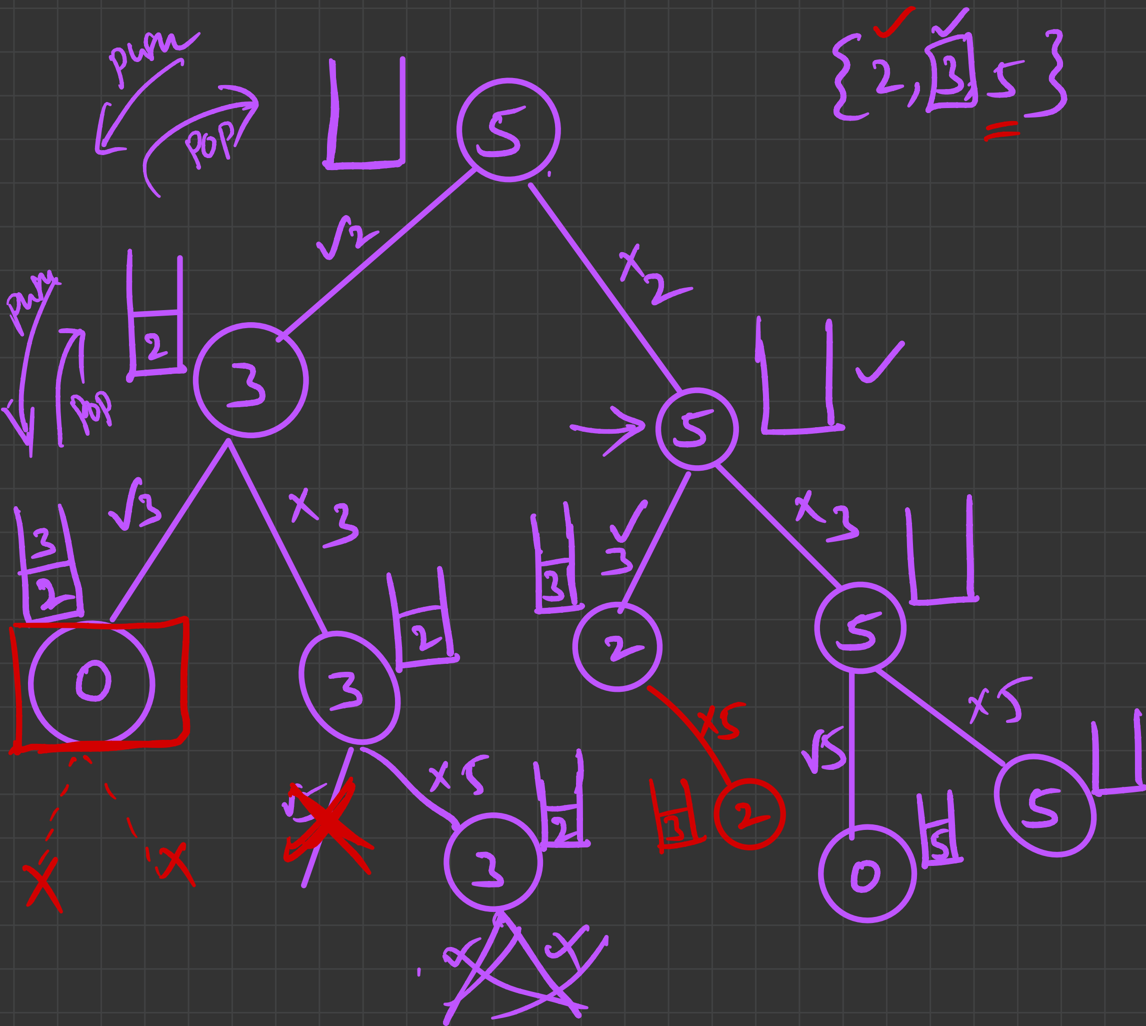
→ T/F

→ int

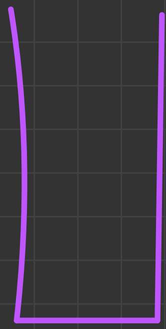
→ int

→ $\{\underbrace{\{2,3\}}_{B}, \underbrace{\{5\}}_{B}\}$

Ans



{2, 3, 5}
push
pop
5
√2
×2
3
push
pop
2
5
√3
×3
√3
×3
0
3
2
5
×5
√5
×5
×5
0
5

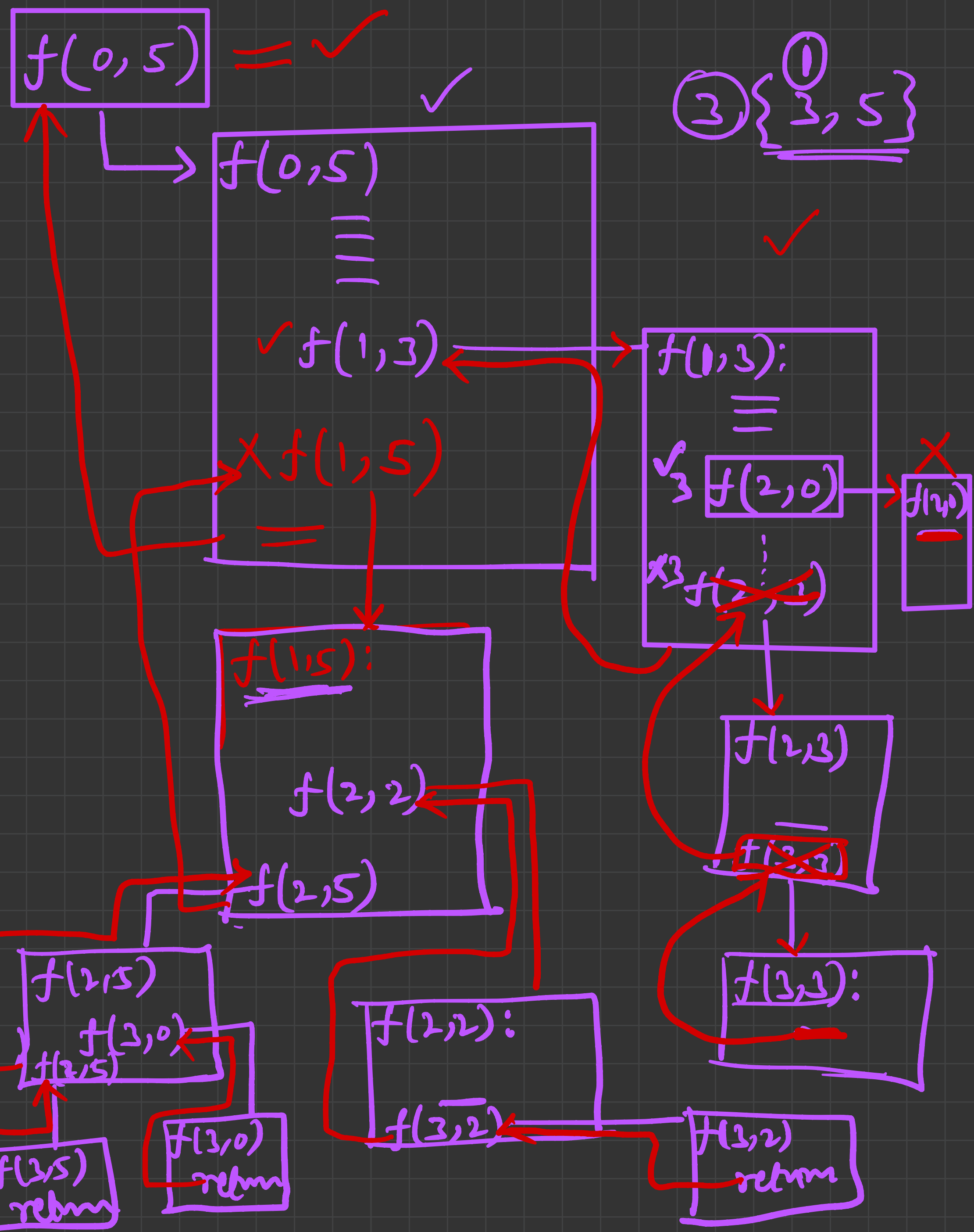


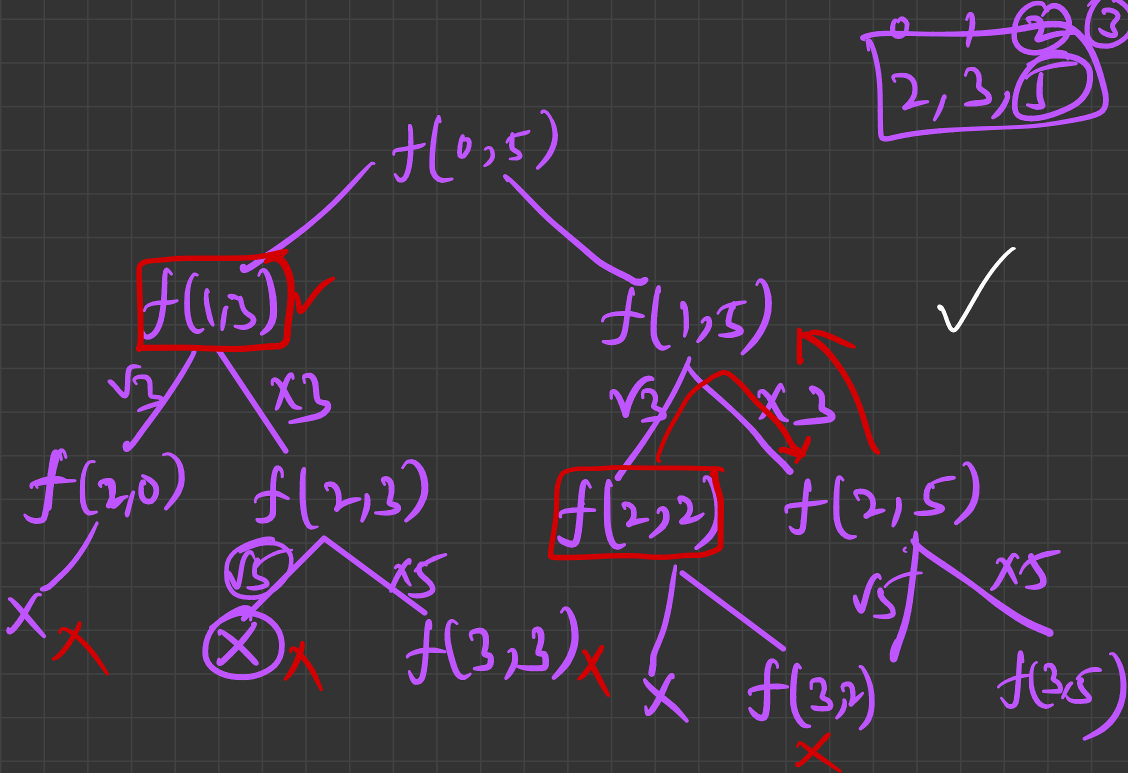
S

```
S.push(5)
S.push(6)
S.pop()
S.push(8)
S.pop()
S.pop()
```

```
f(i, T, A, B, Ans){
  if(T==0):
      Ans.push_back(B)
      return
  if(i ≥ size(A)) return
  if(T ≥ A[i])
      B.push_back(A[i]);
    → f(i+1, T-A[i], A, B, Ans);
      B.pop();
  f(i+1, T, A, B, Ans)
```

Tree ⟶ "call stack" $\{2,3,5\}$





0 1 2 3
2, 3, 1
f(0,5)
f(1,3)
f(1,5)
√3
x3
√3
x3
f(2,0)
f(2,3)
f(2,2)
f(2,5)
X
√5
x5
√5
x5
f(3,3)
X
f(3,2)
f(3,5)

DAY-4

- prime numbers!

$1 \leq n \leq 1000$; $0 \leq \underline{\underline{K}} \leq n$

$n = 15, \; K = 2$

$15 = (\underset{P_1}{\_\_} + \underset{P_2}{\_\_})$ $\underline{\underline{K=2}}$

$(\underset{P_1}{\_\_} + \underset{P_2}{\_\_} + \underset{P_3}{\_\_})$ Hae

$K=2$; $15 = \underline{2} + \underline{13}$

$\boxed{f(\overset{n}{15}, \overset{K}{2}) = 1}$, $f(24, 2)$

$2, 3, 5, 7, \underline{11}, \underline{\underline{13}}, 17, 19, 23$

$f(24,2) = n\{11+13, \; \underline{19+5}, \; 7+17\}$

$f(24,2) = 3$

$f(24, 3) = 2$

$1 \le n \le 1000$

$1 \le k \le n$

$f(\underline{7, 1}) = 1$

$f(\underline{7, 2}) = 1$

$f(\underline{7, 3}) = 0$

$n \leftarrow$

$k \leftarrow$

Ans: $f(n, k) = \underline{\underline{?}}$

1) • we have already solved this?

2) • We haven't solved ?

$$\boxed{\checkmark S = \{ \overset{2}{\checkmark}\ \overset{3}{\checkmark}\ \overset{5}{\checkmark}\ \checkmark\ \checkmark \checkmark \checkmark - - - - < n \}}$$

$K =$ "length"

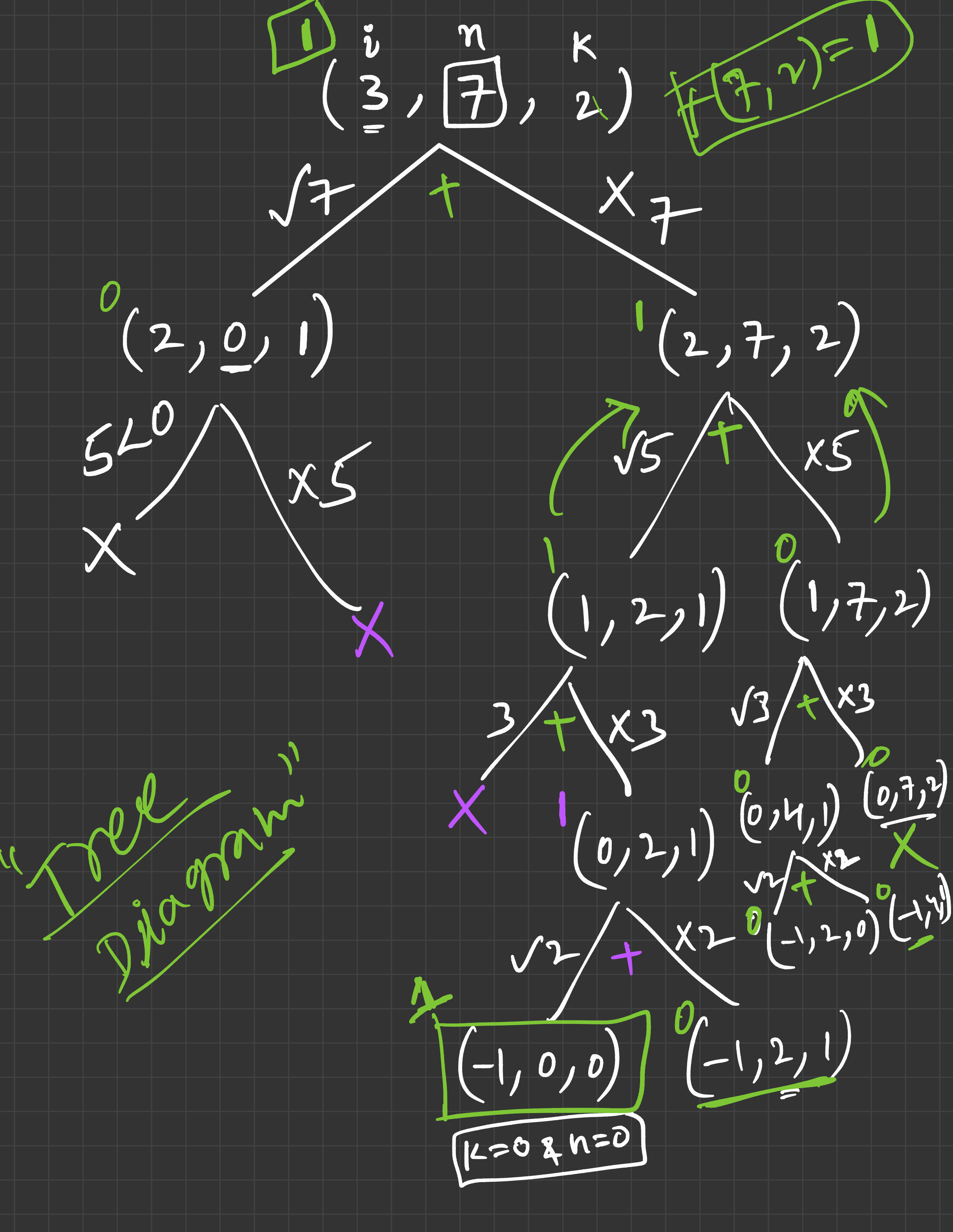
Target sum $\rightarrow \boxed{n}$

$$f(\overset{\checkmark}{i}, \overset{\checkmark}{n}, \overset{\checkmark}{K})$$

$n = 7$

$$A = \{\underset{0}{2}, \underset{1}{3}, \underset{2}{5}, \underset{3}{7}\}$$

$\boxed{K = 2}$



1
i n k
(3, 7, 2)
f(7, 2) = 1
√7
+
×7
0 (2, 0, 1)
1 (2, 7, 2)
5<0
X
×5
X
√5
+
×5
1 (1, 2, 1)
0 (1, 7, 2)
3
+
×3
X
1 (0, 2, 1)
√3
+
×3
0 (0, 4, 1)
(0, 7, 2)
X
√2
+
×2
0 (-1, 2, 0)
0 (-1, 4, 1)
√2
+
×2
1 (-1, 0, 0)
0 (-1, 2, 1)
k=0 & n=0
"Tree Diagram"

```
f(i, n, k):
    if(n==0 && k==0) return 1
    if(n==0 || i<0) return 0
    if( k>i+1) return 0
  if(A[i] <= n):
  Take = f(i-1, n-A[i], k-1);

  NotTake = f(i-1, n, k)

  return Take + NotTake;
```

```
f(i, n, k):
    if(n==0 && k==0) return
    if(n==0 || i<0) return 0
    if( k > i+1) return 0
    if(DP[i][n][k] != -1)
                    return DP[i][n][k]

  if(A[i] ≤ n):
  Take = f(i-1, n-A[i], k-1);

  NotTake = f(i-1, n, k)

return DP[i][n][k] = Take + NotTake;
```

1/0 Knapsack

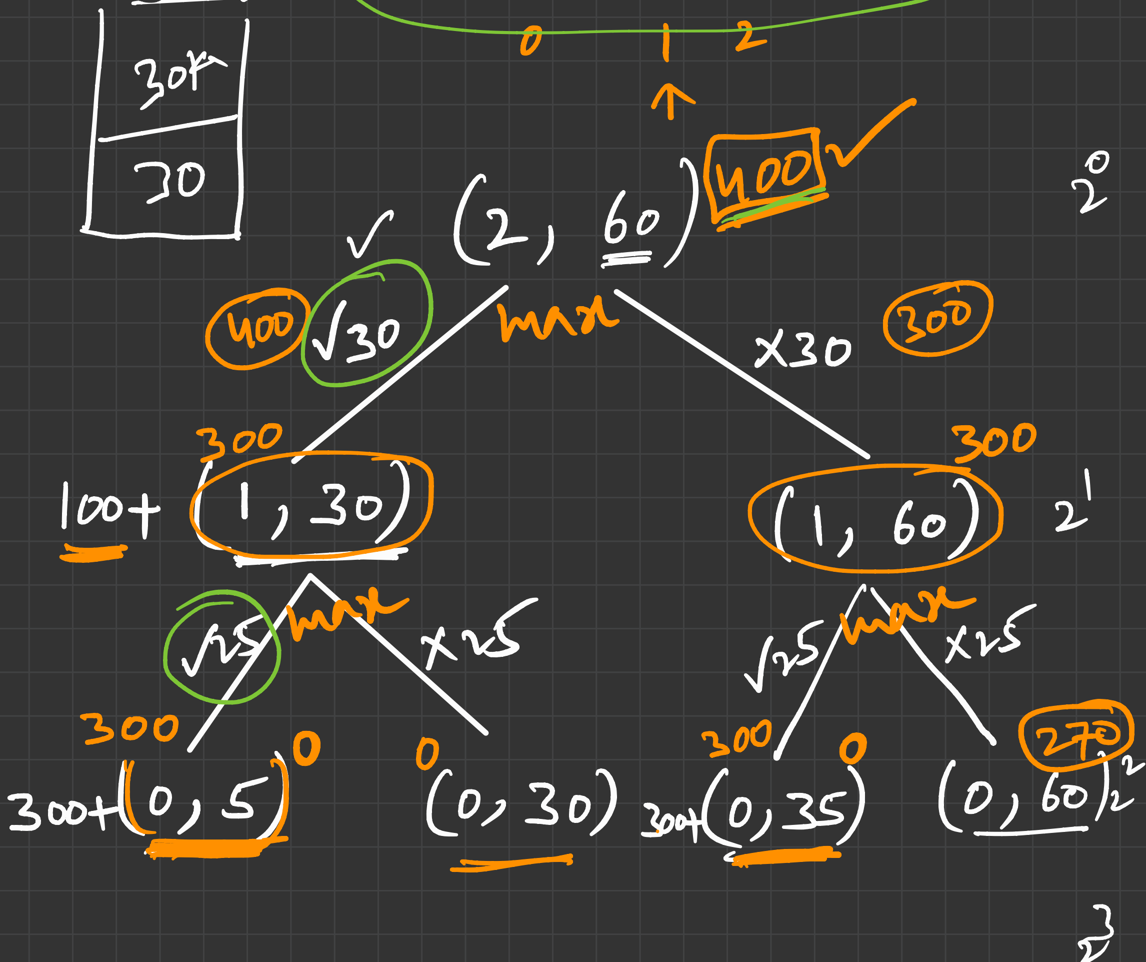
"Bag"

W =	50	25	30
P =	270	300	100
	0	1	2

check

optimize price

60 kg



max

"Smallest problem"

w 4

0

50

270

0

270

max

```
f(i, W):
    if(i==0):
        if(w[0] ≤ W) return P[i]
        else return 0

    Take = -INF
    if(w[i] ≤ W):
        Take = P[i] + f(i-1, W-w[i])

    notTake = f(i-1, W)
    return max(Take, notTake)
```

n=30

30
2

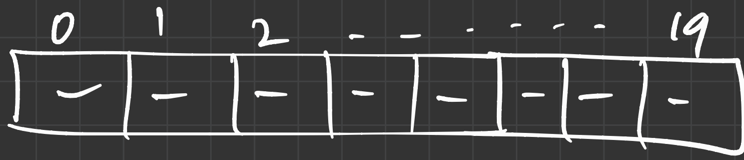
$f(n-1, W)$

w = 

W ← bag capacity.

p = 

```
f(i, W):
  if(i==0):
      if(w[0] ≤ W) return P[i]
      else return 0
  if(DP[i][W] != -1)   return DP[i][W]

  Take = -INF
  if(w[i] ≤ W):
      Take = P[i] + f(i-1, W-w[i])

  notTake = f(i-1, W)

 return DP[i][W] = max(Take, notTake)
```



$W = \underline{100}$ kg ✓

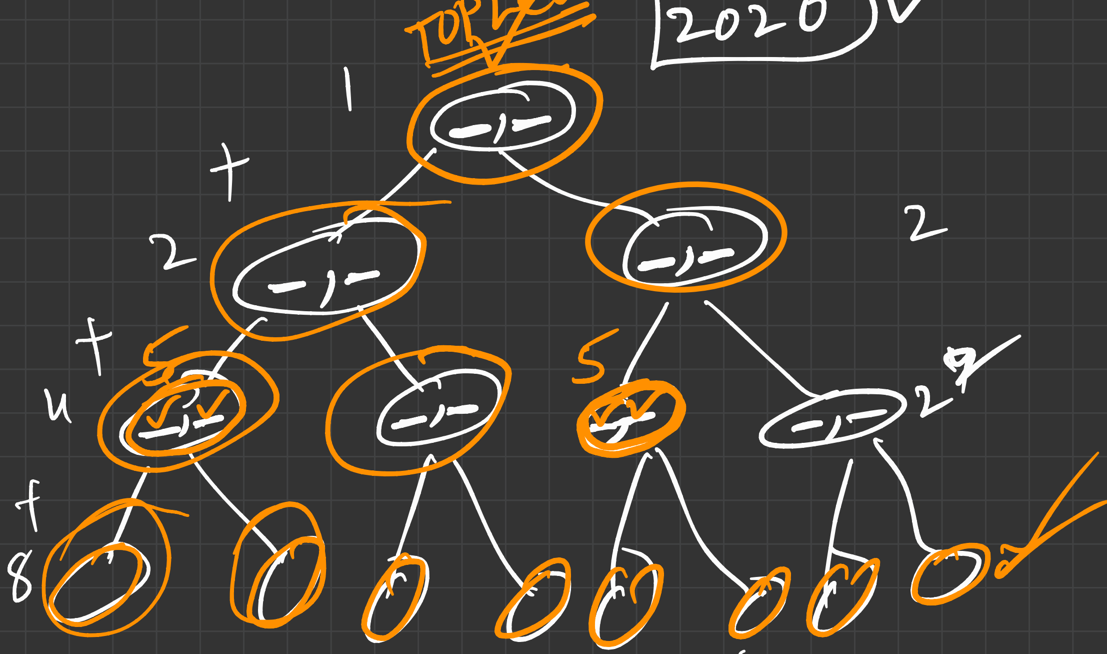
(i, w)

$\underline{(0-19)} \times (0-\underline{100})$

$(\_, \_)$

$(20) \times (101)$

$\boxed{2020}$ ✓



$$1+2+2^2+2^3+\cdots+2^{20}=\boxed{2^{21}-1}$$

$$2020 \ll 2^{21}-1$$

TOP-Down

Ex-1

TRY!!

	0	1	2	3	4	5
W	60	20	50	25	30	70
P	100	110	70	200	150	60
	3	2	1	4	2	5

✓

W = 400

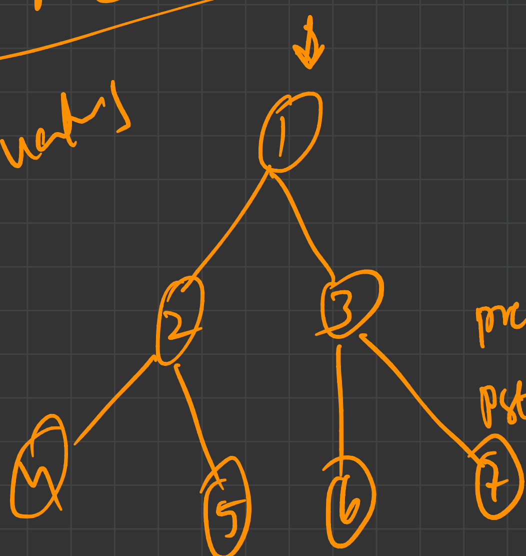
"optimize"

TRY!!!

Tree Traversal

Data structures

"Tools"



LCR

pre: CLR

post: LRC

Inorder: 4, 2, 5, 1, 6, 3, 7 ✓

"Recursive Thinking"

Practice!!

Ex-2 (1-2) Hours

TRY!!

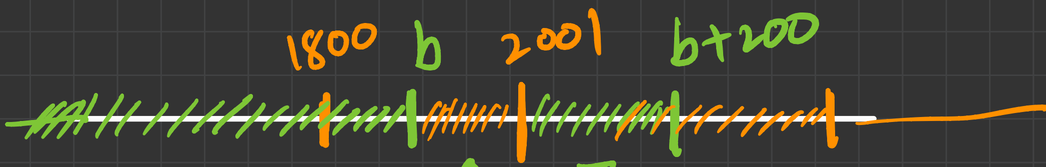
P =	570	230	1500	2010	1600
f =	4	2	3	4	5

$B = 3000$

Buy items such that $\max(f)$

$\max(f) = ?$

$? > 2000$ Discount 200



S_t $\boxed{P}$ ✓

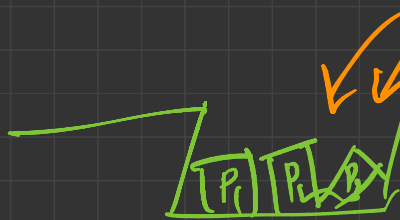
b+200

b = 1900 ✓

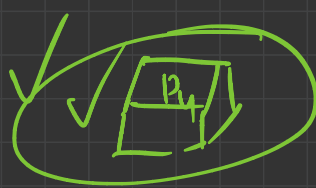
$$\begin{array}{r} 1900 \\ 200 \\ \hline 2100 \end{array}$$

1950	30	10	✓	✓	✓
5	2	1			

Ans : 3



S



$S + P_4 < b$

$\boxed{p_5}$ $\boxed{s+p_5 > b}$

$s+p_5 \le 2001$

$s+p_5 \le b+200$

1900 + 200

200

> 2000

$s = 1950 + \_\_ + \_\_ + \_\_$

$i \rightarrow$ 1 2 3 4 5

$\boxed{>2000}$

2000 1600

5 2 1

$s \le 2100$

>2000

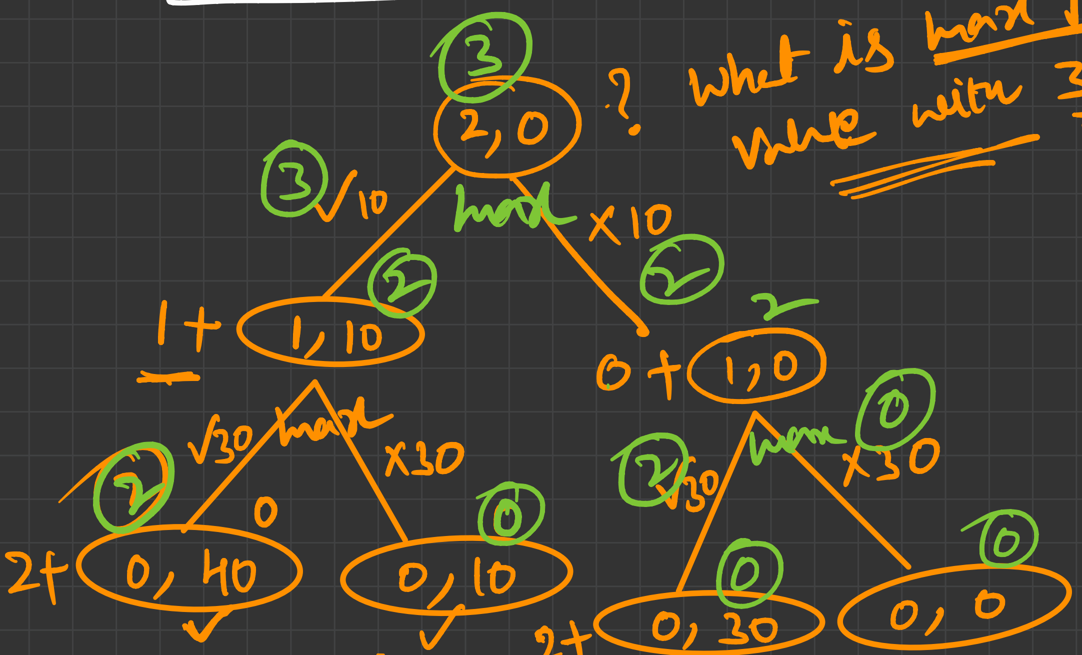
$s \le b+200$

? ?

$b = 1900$

0	1	2
1950	30	10
5	2	1

What is max fav value with 3 ing



$40 + 1950 = 1990$

$S + P_0$

$S \leq b$

$b = 1900$

$S + P_0 > b$

$S + P_0 < 2001$

b

f_0 0

0, 0

✓ p ✓✓

✓ p ✗

✓ — ?

0

p

f_0

S

0, S

0

p

f_0

S

$S > b$

b pick $S + p \leq b$ ✓✓

p pick $S + p > b$ && $S + p < b + 200$ && $S + p \geq 2001$ ✗ ✓ ✓

Not pick $S \le b$ ✓

$S > b$

✓✓✓

$S > b$

$S + P > b$ && $S + P \le 2000$

$S \le b$

$S > b$

Walmart

$\boxed{S}$

$S \le b + 200$ &&

$S \ge 2001$

Kart ✓

$\boxed{S + P}$ ✓

$S + P \le b$ ✓

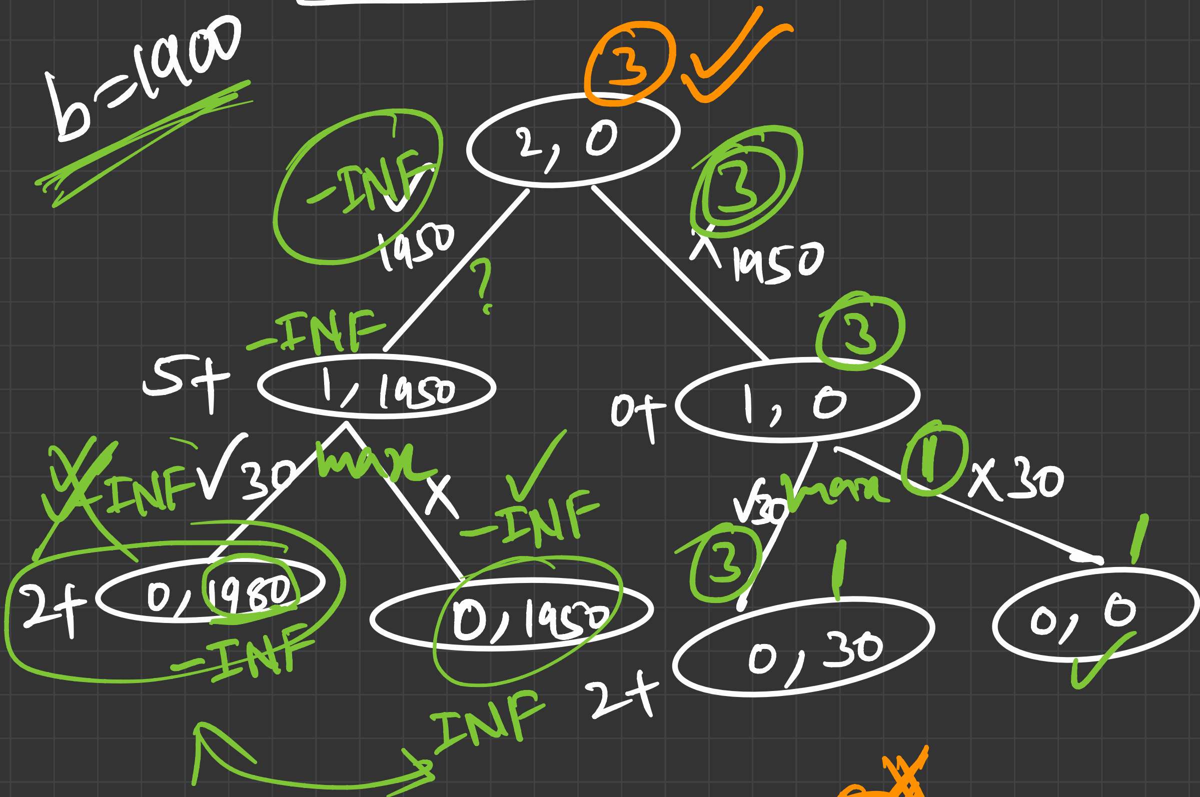
✓ $S + P > b$ && $S + P < b + 200$ ✓

$S + P \ge 2001$

$b > 1800$ ✓

b = 1900

0	1	2
10	30	1950
1	2	5



(0, 1980) → 1980 + ~~10~~ = 1990 X

NotPick

$$\boxed{f(n-1, 0)}$$

```
f(i, s):
  if(i==0){
    if(s+p[0] <= b || b>1800 && s+p[0] >= 2001
                       && s+p[0] <= b+200)
       return F[0]
    if(b>1800 && s <= b+200 && s >= 2001)
       return 0
    if(s>b) return -INF
    return 0
  }
  Take = -INF
  if(p[i] <= b || s+p[i] <= b+200 &&
                  b>1800)

     Take = F[i] + f(i-1, s+p[i])

  notTake = f(i-1, s)

  return max(Take, notTake)
```

```
for(int i=0; i<n; i++){
    cout<< i <<"\n";
}
```

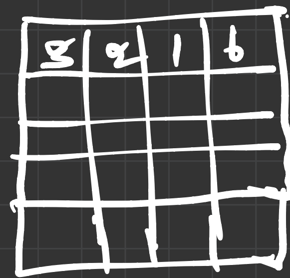
```
for(int i=0; i<5; i++){
    for (int j=2; j<4+2; j++ ){
        cout<<A[i][j]<<" ";
    }                      A[i][j%4] ✓
    cout<<"\n";
}
```

int A[10][5]

$$
\begin{array}{cc|ccccc}
 & & & & j & & \\
 & & 0 & 1 & 2 & 3 & 4 \\
 & 0 & -1 & 5 & 2 & 6 & 1 \\
 & 1 & & & & & \\
i & 2 & 3 & 2 & 0 & 1 & -5 \\
 & \vdots & \cdot & \cdot & \cdot & \cdot & \cdot \\
 & 9 & 6 & 2 & 0 & 6 & -7
\end{array}
$$

10x5

$A[5][4]$



5×4

$$
\begin{array}{c}
j = 0 \quad 1 \quad 2 \quad 3 \\
0 \quad 1 \quad 2 \quad 3
\end{array}
$$

$$
A = \begin{bmatrix} 3 & 0 & 1 & 6 \\ -1 & 0 & 0 & 1 \\ 3 & 6 & 0 & 5 \\ 3 & 0 & 2 & -1 \\ 6 & 2 & 5 & 1 \end{bmatrix}
$$

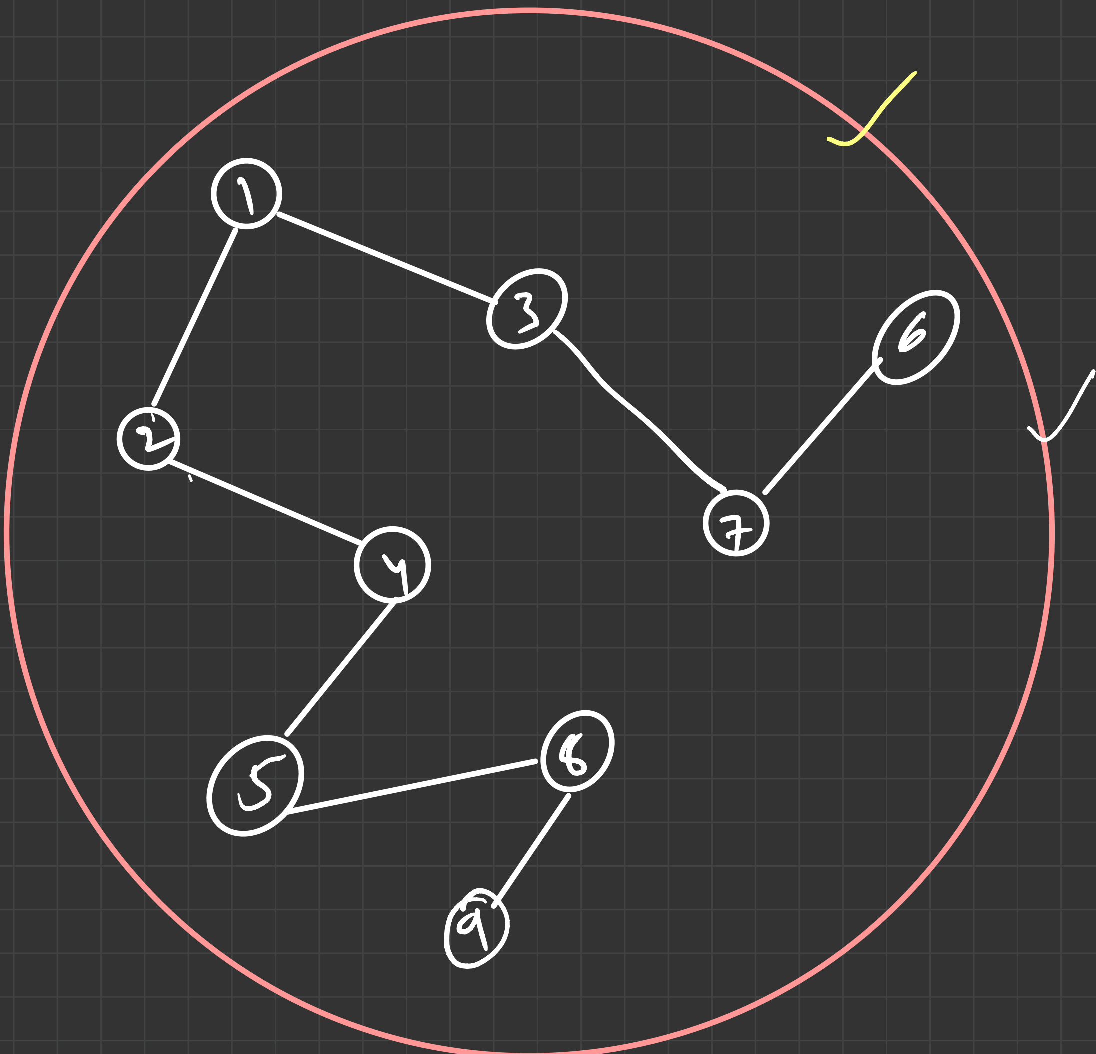
1	6	3	2
0	1	-1	2
1	5	3	6
2	-1	3	1
5	1	6	2

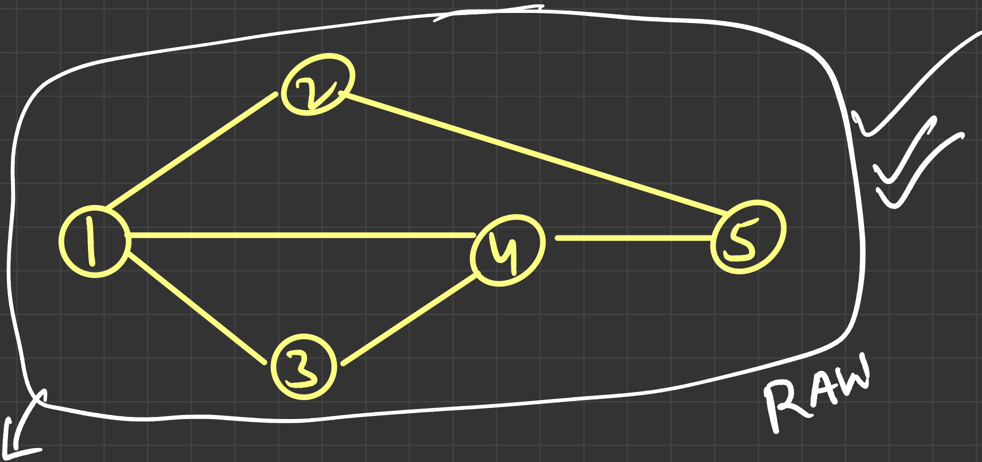
1	6	3	
-1	1		
5	3	6	
3	-1	2	
5	1	6	2

Exercise!!

Graphs

- cyclic / acyclic
- Adj Matrix ✓
- Adj List ✓





RAW

n	m
5	6
1	2
4	1
1	3
2	5
4	5
4	3

Adj List

1 : {2, 3, 4}

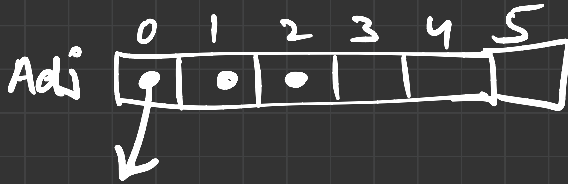
2 : {1, 5}

3 : {1, 4}

4 : {1, 3, 5}

5 : {2, 4}

vector<int> Adj[6]



Adj[0] = [ ]

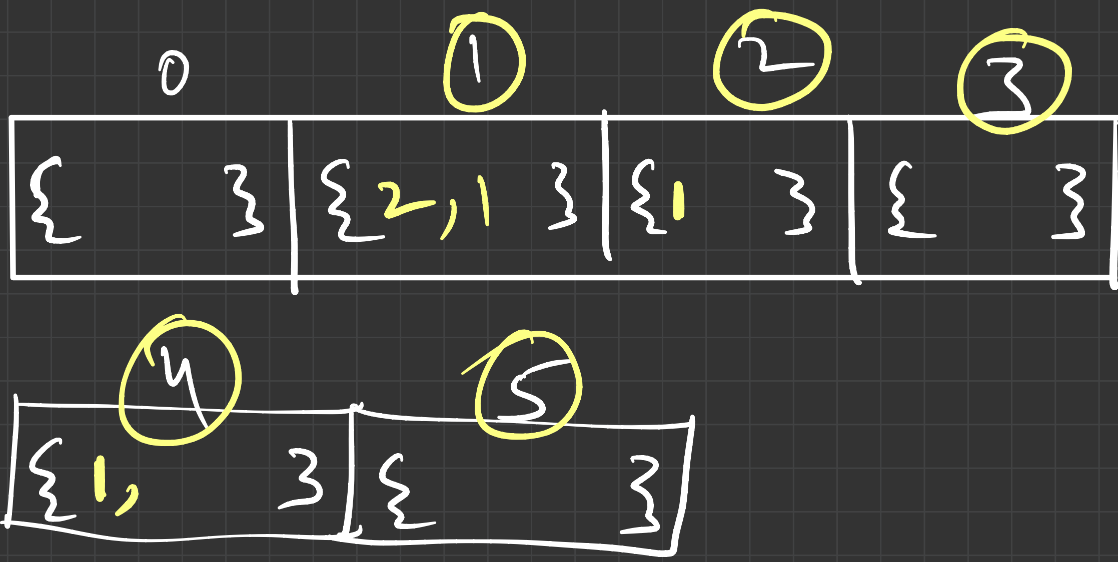
Adj[1] = [ ]

Adj[2] = [ ]

Adj[3] = [ ]

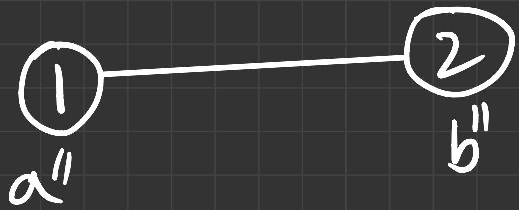
Adj[4] = [ ]

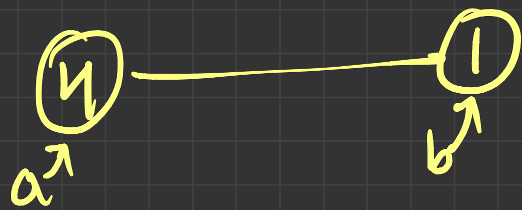
Adj[5] = [ ]



✓ adj[a].push_back(b) ✓

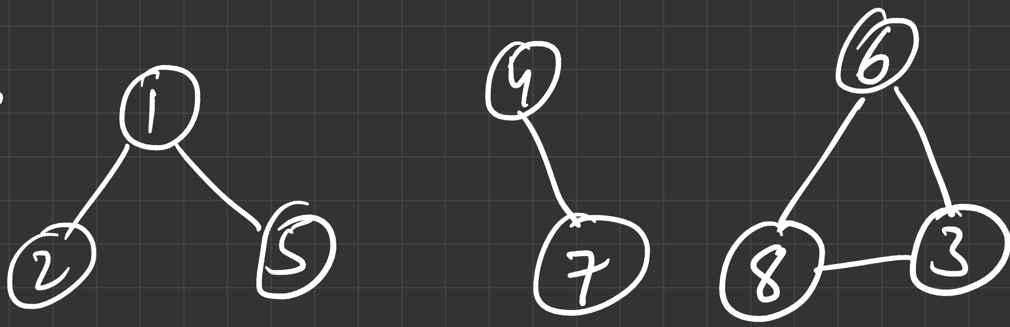
✓ adj[b].push_back(a)





Basic problems:

1.



Explicit graph

Adj[0] = { }

Adj[1] = {2, 5}

Adj[2] = {1}

Adj[3] = {8, 6}

Adj[4] = {7}

Adj[5] = {1}

Adj[6] = {3, 8}

Adj[7] = {4}

Adj[8] = {6, 3}

$n \quad m$

$8 \quad 6$

How many components?

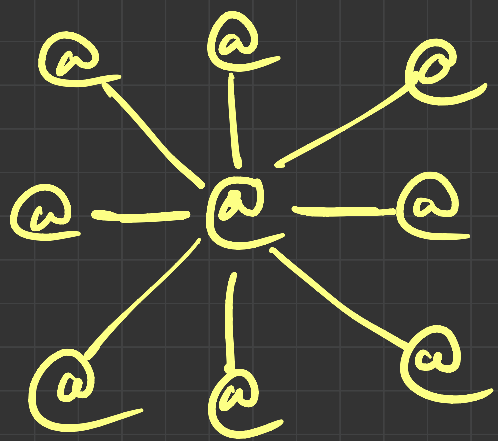
②

	0	1	2	3
0	@	@	•	•
1	@	•	•	•
2	•	•	@	@
3	@	@	•	•
4	•	•	@	•

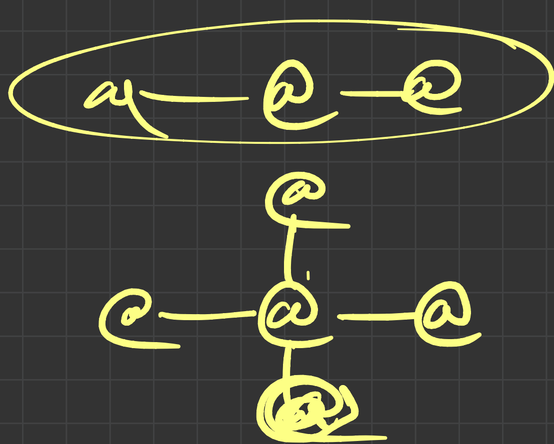
@ ✓

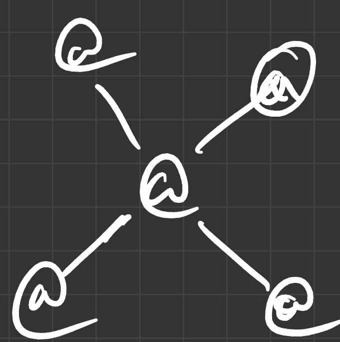
• X

@



Implicit graph





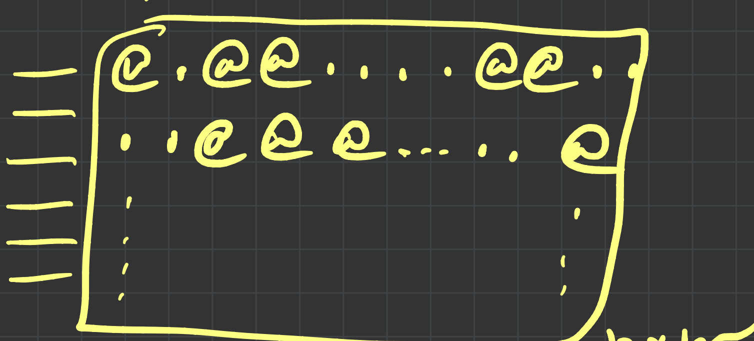
.	@	.	@	.
@	.	@	.	@
.	@	.	@	.
@	.	@	.	@

100X100

→ Rotation

→ Ellipsiod

I/P: n, m



O/P: $\boxed{}$

$\boxed{\text{DFS}}$

	0	1	2	3
0	@	@	.	@
1	@	.	.	.
2	.	.	@	@
3	@	@	.	.
4	.	.	@	.

6

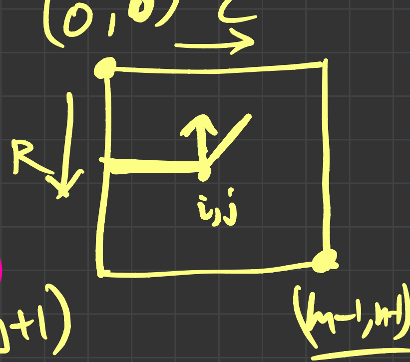
	0	1	2	3
0	1	1	0	0
1	1	0	0	0
2	0	0	1	1
3	1	1	0	0
4	0	0	1	0

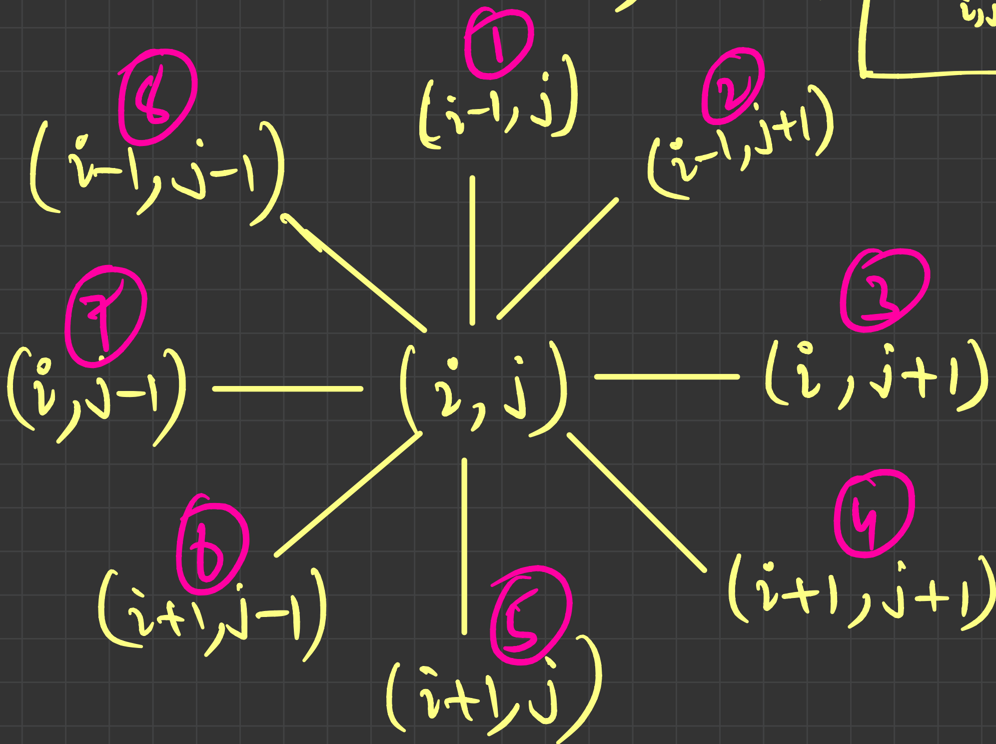
VIS

```
m, n ←
grid ←
vis[m][n] = {0}

for(i=0; i<m; i++){
    for(j=0; j<n; j++){
        if(grid[i][j]=='@' && vis[i][j]
                                  ==0)
        {  DFS(i,j)
           count++
        }
    }
}
print(count);
```

DFS(i, j){

vis[i][j] = 1;





$i + \{-1, -1, 0, 1, 1, 1, 0, -1\}$

$j + \{0, 1, 1, 1, 0, -1, -1, -1\}$

```
DFS ( i, j) {

    Vis [i][j] = 1;

    row[ ] = {-1, -1, 0, 1, 1, 1, 0, -1}
    cal[ ] = {0, 1, 1, 1, 0, -1, -1, -1}

    for (k=0; k<8; k++) {
        new_i = i + row[k];
        new_j = j + cal[k];
        if (Grid [new_i][new_j] == '@'
            && vis[new_i][new_j] == 0)
            DFS(new_i, new_j)
    }

} // End of DFS
```

Ex1: "cylindrical CC"

Ex2: "No. of @ in Largest compnent"

	0	1	2	3
0	.	@	@	@
1	@	@	.	@
2	.	.	.	.
3	@	@	@	@

✓

Q	
~~(3,3)~~	3
~~(3,2)~~	2
~~(3,1)~~	1
~~(3,0)~~	0
~~(1,3)~~	2
~~(2,3)~~	2
~~(1,0)~~	1
~~(1,1)~~	1
~~(0,2)~~	1
~~(0,1)~~	0

Q

cnt++

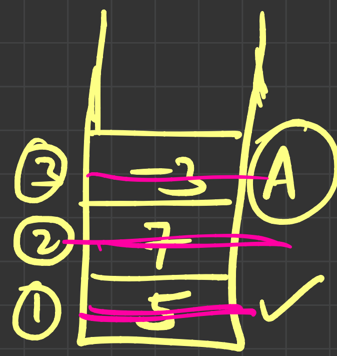
0	1	1	1
1	1	0	1
0	0	0	0
1	1	1	1

✓

```
for( i=0;  i<m ; i++){
    for(j=0; j<n ; j++){
        if(grid[i][j] == '0' &&
            vis[i][j] == 0){
            count++;
            BFS(i, j);
    }
```

```
BFS (i, j) {
    vis[i][j] = 1;
    row[] = {-1, -1, 0, 1, 1, 1, 0, -1}
    cal[] = {0, 1, 1, 1, 0, -1, -1, -1}

    queue<pair<int, int>> Q;
    Q.push({i, j})
    r = Q.front().first ;
    c = Q.front().second ;
    Q.pop( );
->  while(!Q.empty()){
    for(k=0; k<8; k++){
        newrow = r + row[k]
        newcal = c + cal[k]
        if(grid[newrow][newcal]=='@'
            && vis[newrow][newcal]==0)
```

```
Q.push({newrow, newcol})
    }
}
```




```
queue<int> A;
A.push(5)
A.push(7)
A.push(-3)
```

$A.front() \longrightarrow 5$

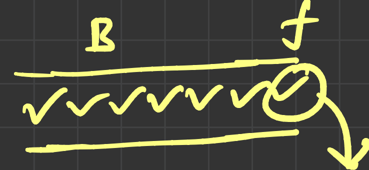
```
cout << A.front();
A.pop();
```

`cout << A.front();` $\longrightarrow 7$

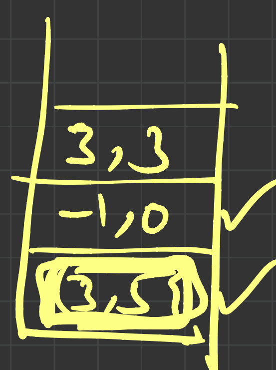
```
A.pop();
```

`cout << A.front();` $\longrightarrow -3$

```
A.pop();
```



```
pair<int, int> P = {2,3}

int A = 5;

cout << P.first;  → 2
cout << P.second; → 3

queue< pair<int,int>> Q

Q.push({3,5})
Q.push({-1,0})
Q.push(3,3)

Q.front().first  ⇒ 3
Q.front().second ⇒ 5
```



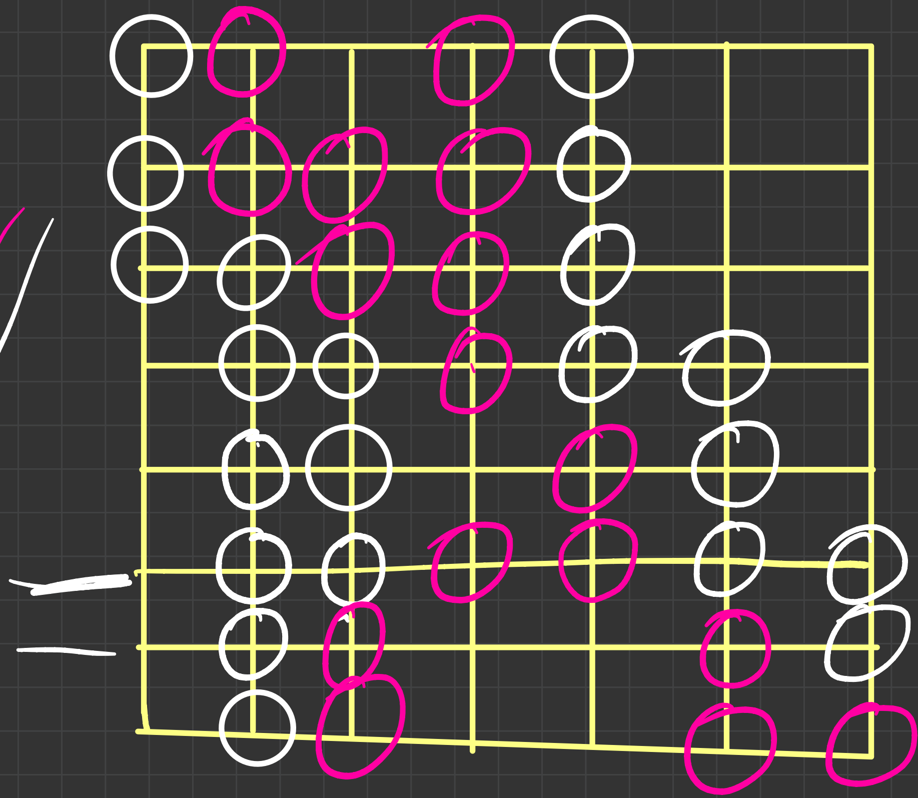
3

	0	1	2	3
0	@	@	•	•
1	@	•	•	•
2	•	•	@	@
3	@	@	•	•
4	•	•	@	•

"number of nodes in the largest component"

4

"GO"



Input

```
W B · B W · ·
W B B B W · ·
W W B B W · ·
· W W B W W ·
· W W · B W ·
· W W B B W W
· W B · · B W
· W B · · B B
```

Output

W
B
D

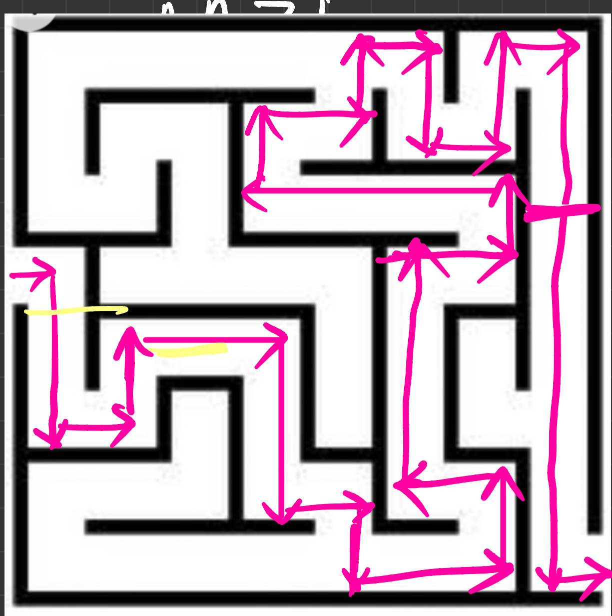
"Traversal"

BFS

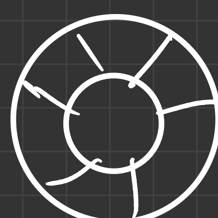
DFS

"Intuition"

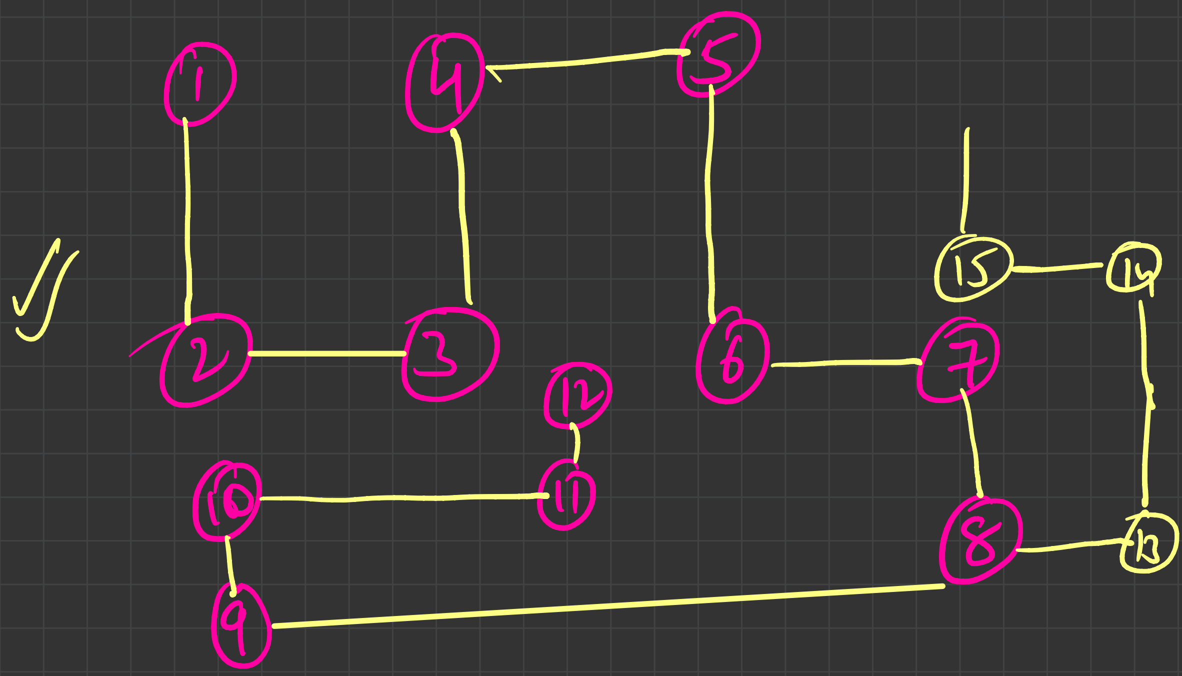
DFS



- UP ↑
- Left ←
- Right →
- Down ↓



⇓ Graph



n m

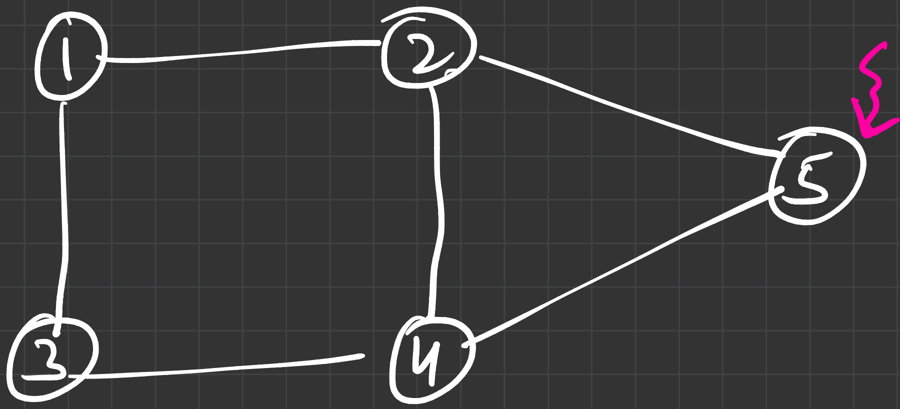
✓ ✓

✓ ✓

✓ ✓

						1
					0	1
0	0				0	1
1	0	0			0	1
1	1	1	1	1	0	1
1	1	1	0	1	0	1
0	0	0	0	1	0	1
1	1	1	1	1	0	1

$m \times n$



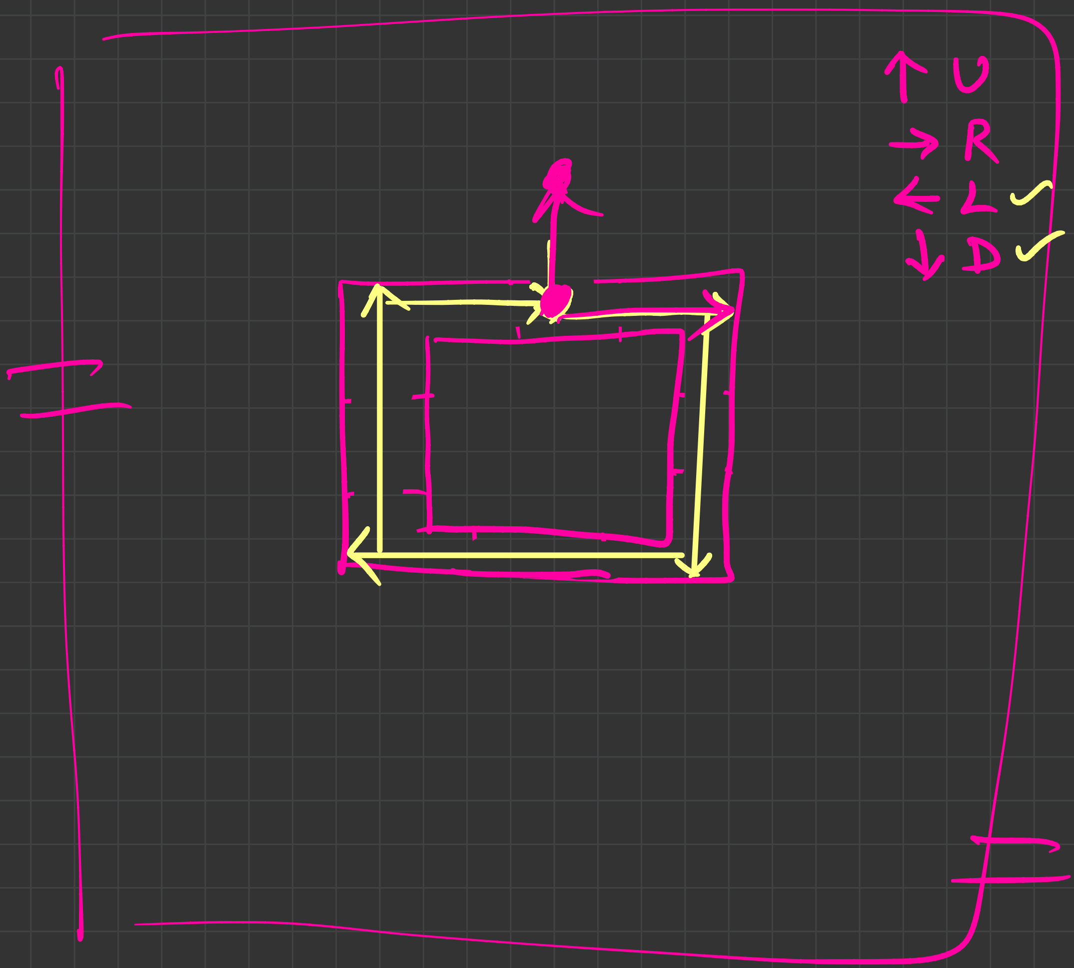
DFS: 1, 3, 4, 5, 2 ✓

1, 2, 4, 3, 5 ✓

1, 2, 5, 4, 3 ✓

BFS: 1, 2, 3, 4, 5 ✓

BFS: 5, 2, 4, 1, 3


↑ U
→ R
← L ✓
↓ D ✓

DFS / BFS

X ✓

$\{0, 1, 2\}$

	0	1	2
0	2^0	2^1	2
1	2^1	2	0
2	2^0	2	0

return $\boxed{-1}$

~~((1,1),2)~~
~~((0,2),2)~~
~~((2,1),1)~~
~~((1,0),1)~~
~~((0,1),1)~~
~~((2,0),0)~~
~~((0,0),0)~~

	0	1	2
0	1	1	1
1	1	1	0
2	1	1	0

2	1	2	0
1	1 ^{2}	0	1
2	1	1	2

2 →

t

+ 1 unit