std :: sort (v.begin() , v.end()) → O(n log n)

STL → optimized & general Algorithm

* Bubble Sort → $O(n^2)$
* Quicksort
* Merge - sort → $O(n \log n)$
* ...

for i
  for j
    arr[i] > arr[j]

$n = 2$ ↔ $O(4)$

$O(n \log n) = O(2 \cdot 1)$

$= O(2)$

$O(n)$, $O(n \log n)$

n ... $n \log n$

1 ... $\log n$

$O(1)$ $O(\lg n)$

$n = 1$ → $O(1)$

→ $O(\log n) = O(2^{\log 1}) = O(0)$

Lebih cepat ($O(\log n)$)

$n = 2$ → $O(1)$

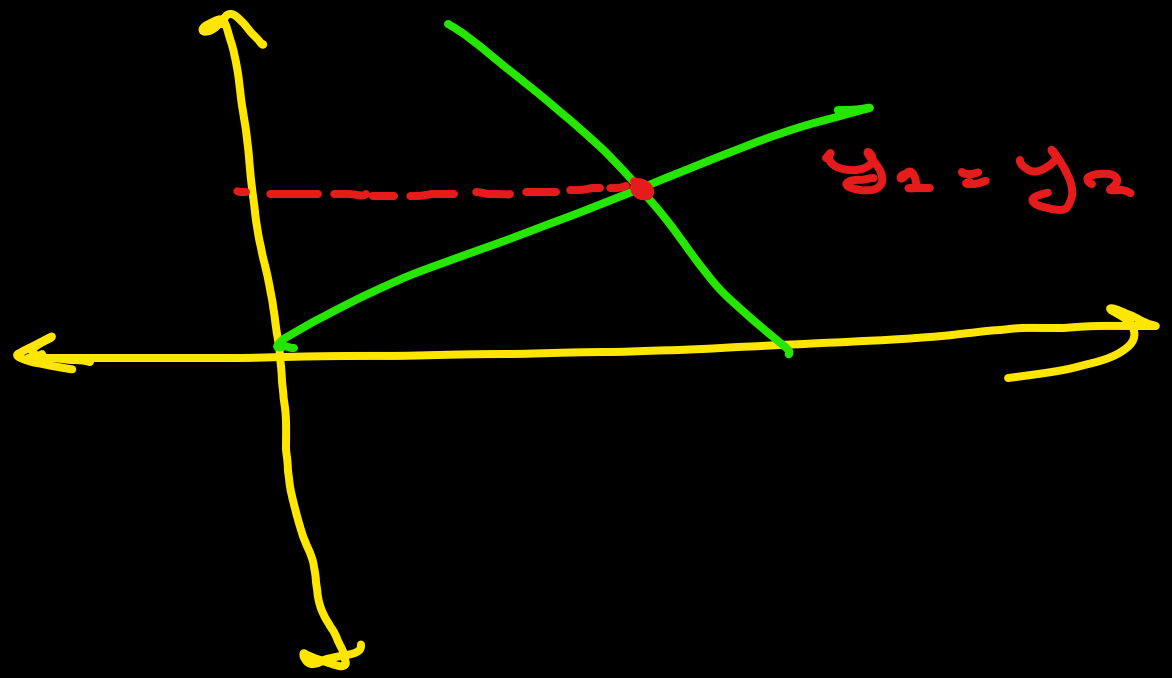
→ $O(\log n) = O(1)$

Sama

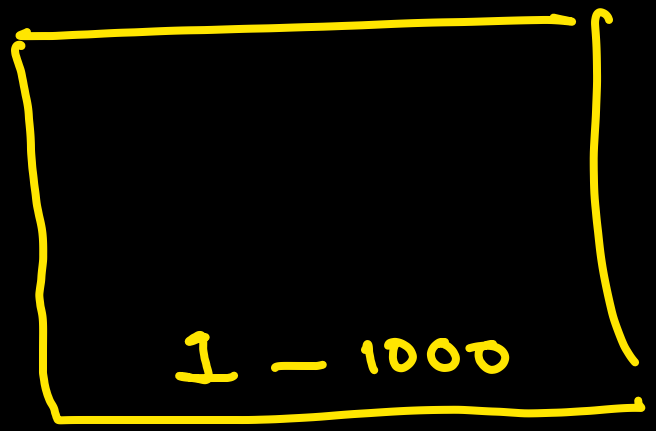
$n = 3$ → $O(1)$

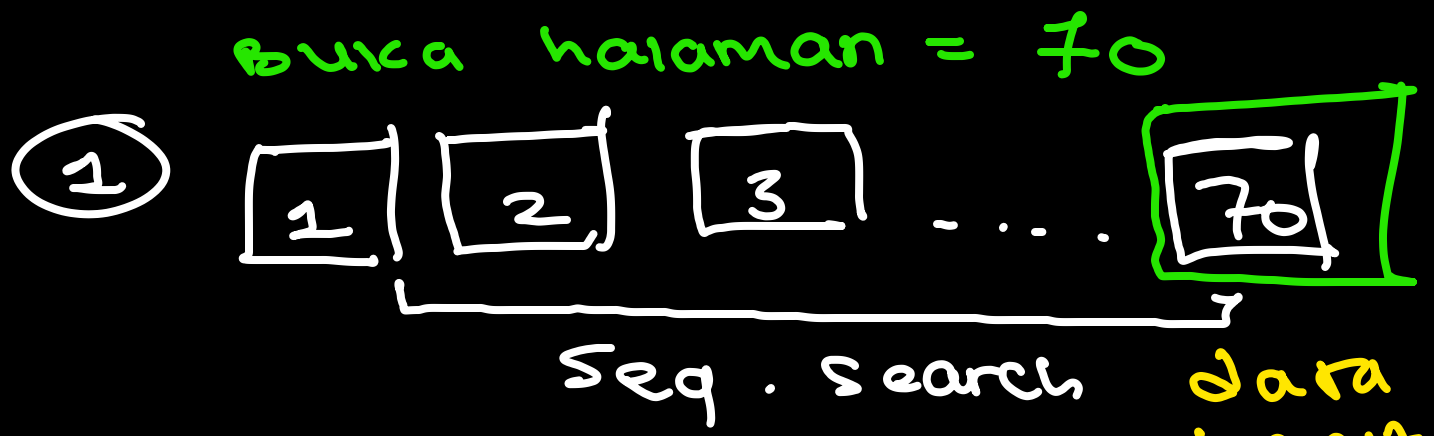
→ $O(\log n) = 2^{\log 3} = 1,\ldots$

Lebih cepat $O(1)$



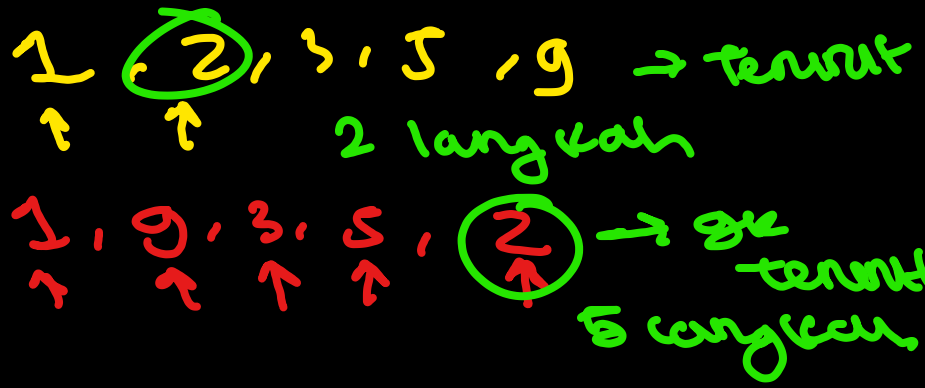
$2 = 2$





1, 9, 3, 5, 2

angka 2



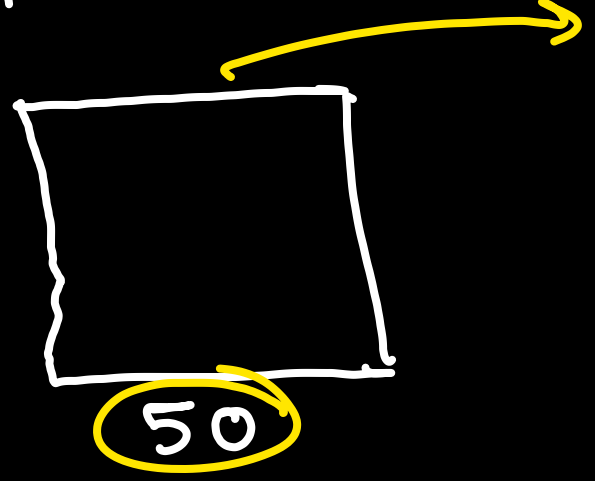

```
arr
find = false;
for data in arr ->
  if(data == cari):
     find = true;
     print("Ketemu")
     break

if(!find):
   print("Tidak ketemu")
```

array berukuran N

* Seq search : O (N)

* Binary Search



70 > 50

search (9)

— urutkan data

1 2 3 5 9

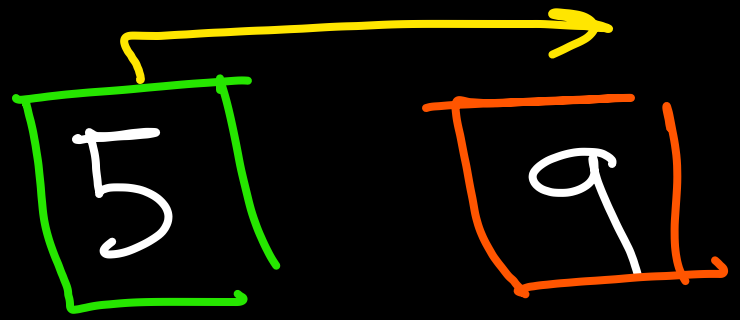
mid = 3

mid - index = 2

9 >= 3

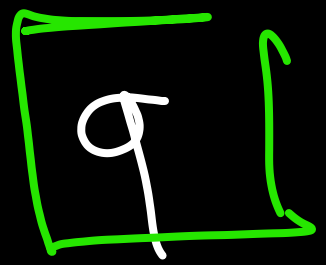
→ kekanan

mid = cari ?



mid

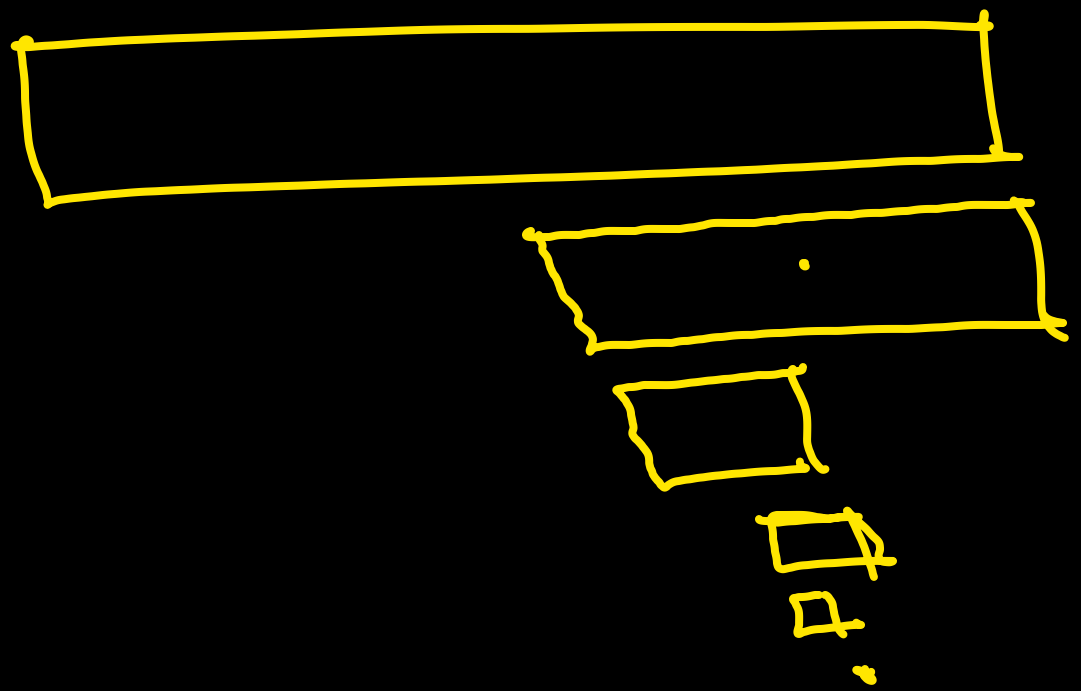
9 > 5



9 = 9

mid = cari

↘ ketemu



/2

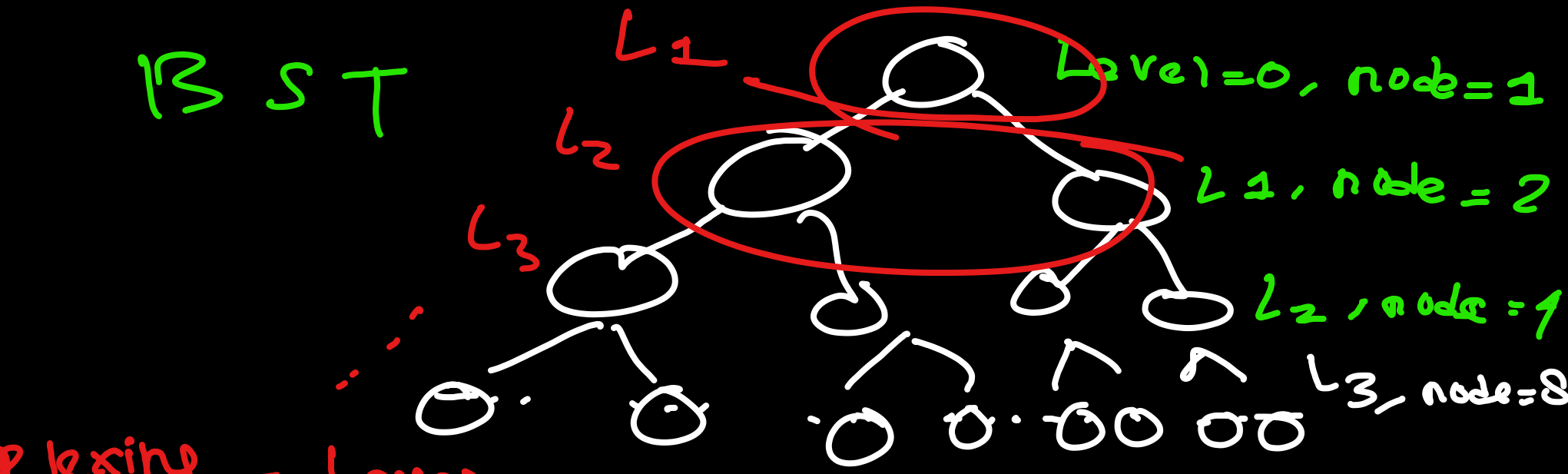
/2

/2

/2

} binary

BST



Complexing = Level

Nodes Level ke-n $= 2^n$

$Nodes = 2^{(n)}$ → nomor level

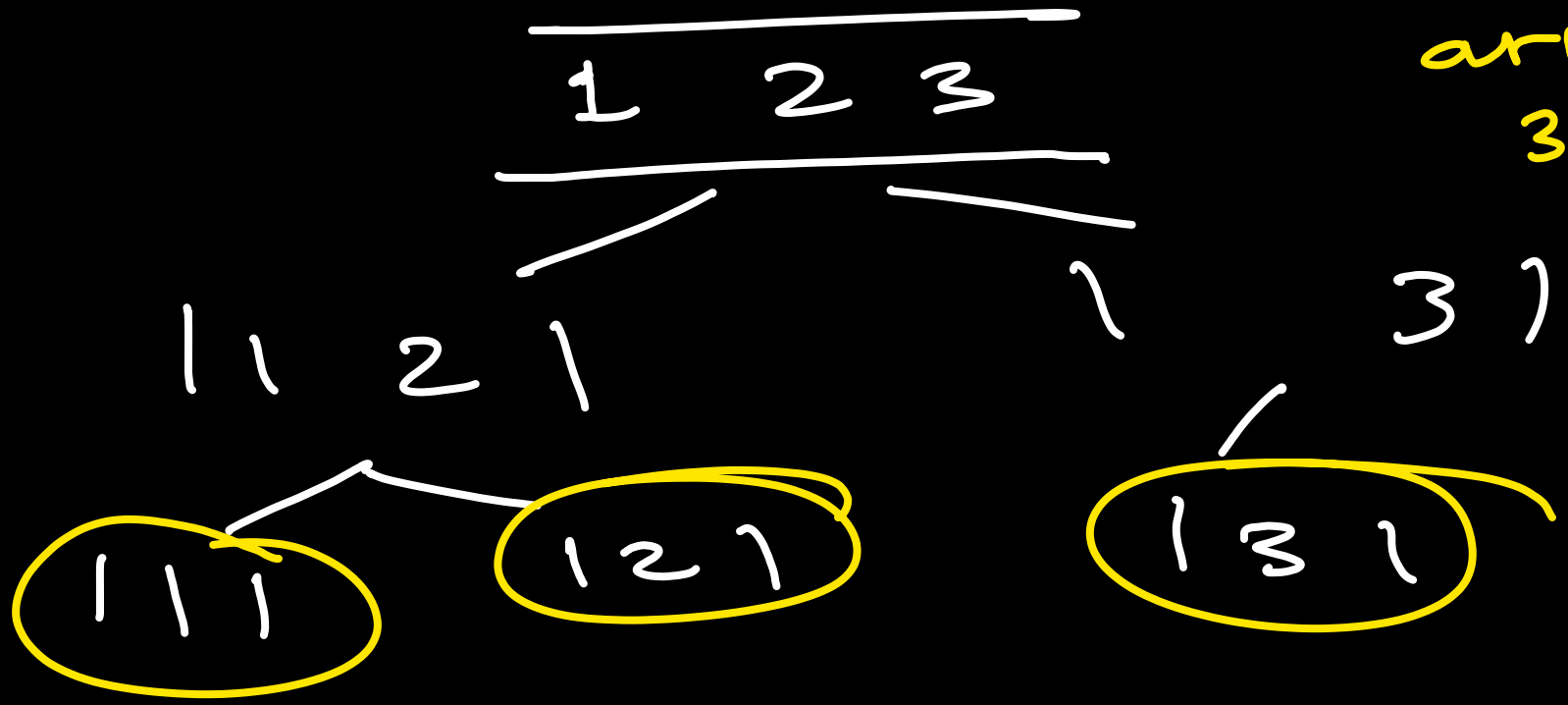
array = N

$n = 2^{\log nodes}$

$Level = 2^{\log nodes}$

$Level = 2^{\log N}$

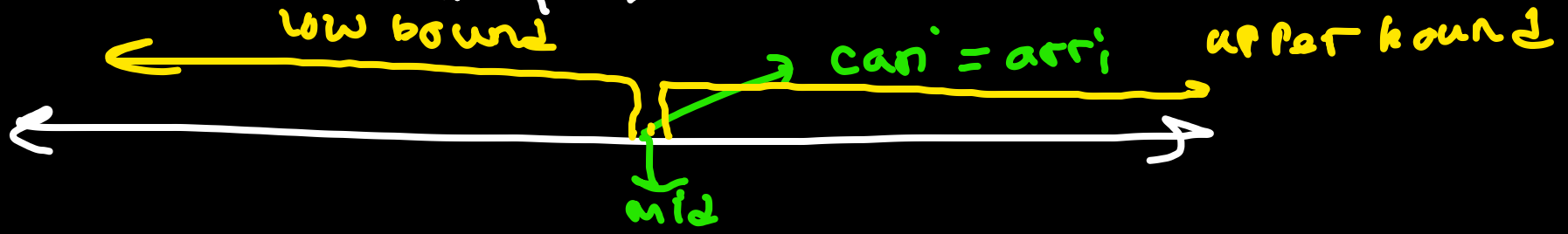
$O(\log N)$



Search → $arr_i = can_i$

Low bound → $arr_i \leq can_i$

Upper bound → $arr_i \geq can_i$



a b c

Kiri = a → low bound (b)
Kanan = c → upper bound (b)

$x = 16$

1 2 3 4 5 6 7 8 9 10 ... 16

L

16

if (arr[mid] == x).

↳ if

N = 5

3! = 6

4! = 24

```
( for ( int i = 1 ; i < N ; i++ ) {
      N *= i
} )
```

i = 1 5 x 1

i = 2 5 x 1 x 2

i = 3 5 x 1 x 2 x 3

i = 4 5 x 1 x 2 x 3 x 4 → 5! = 120

Function Rekursif

U(2020) = U(2019)+1

Umia = Umia sebelum + 1

Umia di tahun ke-n = U(n)

U(n) = U(n-1) + 1

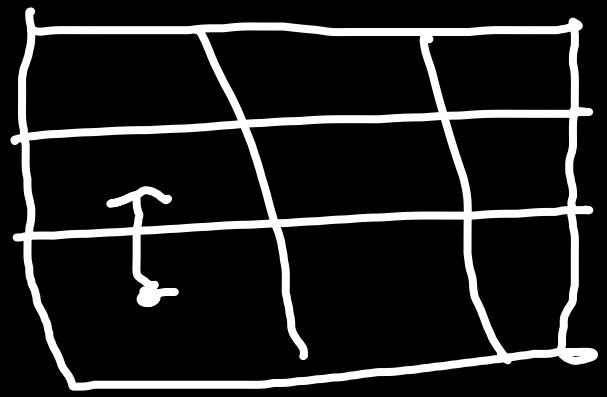
Rekurens

Fungsi memanggil dirinya sendiri

Base case

pos(x,y)

pos-atas = pos(x-1,y)



$$U(2020) = U(2019) + 1$$
$$U(2019) = U(2018) + 1$$
$$U(2018) = \cdots$$

Base case $U(2010) = 1$

$$U(2024) = U(2023) + 1$$
$$U(2023) = U(2022) + 1$$
$$U(2022) = U(2021) + 1$$
$$\cdots$$
$$U(2010) = 1$$

16 → [1, 2, 3, ..., 16]

seq search : $1^2 = 16$? $2^2 = 16$? ... } 4 comparisons

Binary search : $8^2 = 16$?

... $4^2 = 16$? (Yes)

STL sqrt = → sqrt (decomposition) _ binary search

(1) for (1, n) → core seq

(2) while (n/=2) step → seq

(3) Binary search

(4) sqrt decomposition

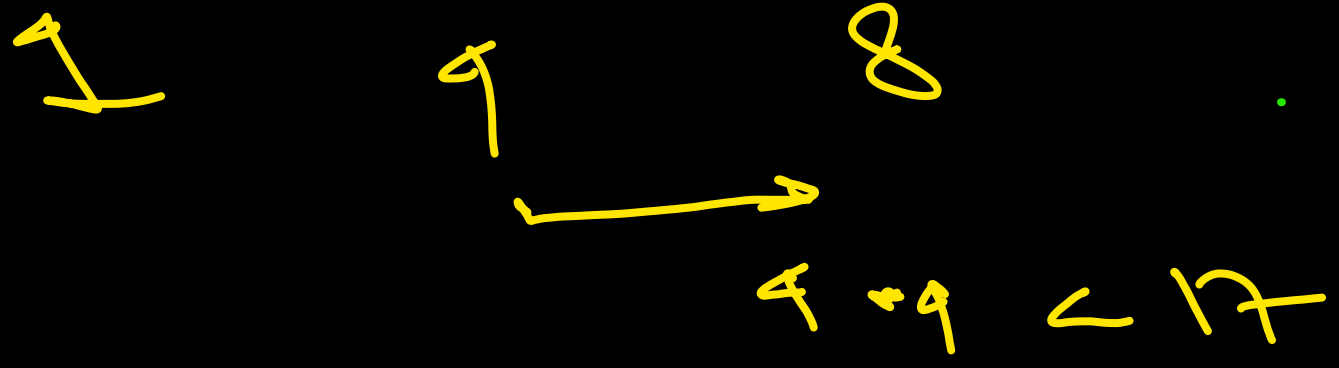
16 ** (0.5) → Binary exp.

17 = [1, ..., 17]

1 | 8 | 17

mid

8 * 8 > 17



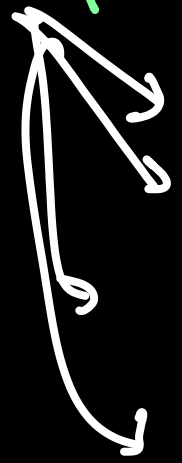
$6 * 6 > 17$

$5 * 5 < 17$?

6

$5 * 5 \rightarrow 4 * 4 = 16$

* PSPC (Problem Solving Paradigm)

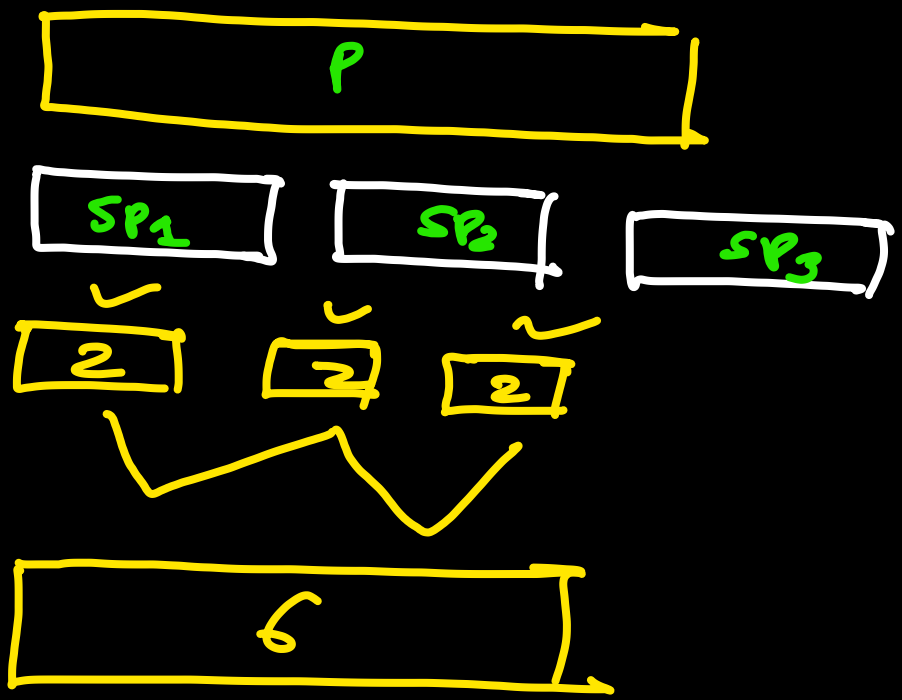


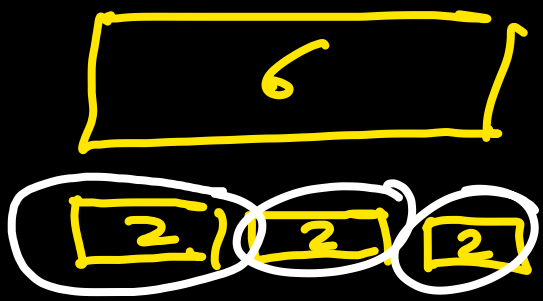
- Greedy →
- Brute-forces
- DnC ↔ Binser
- DP →

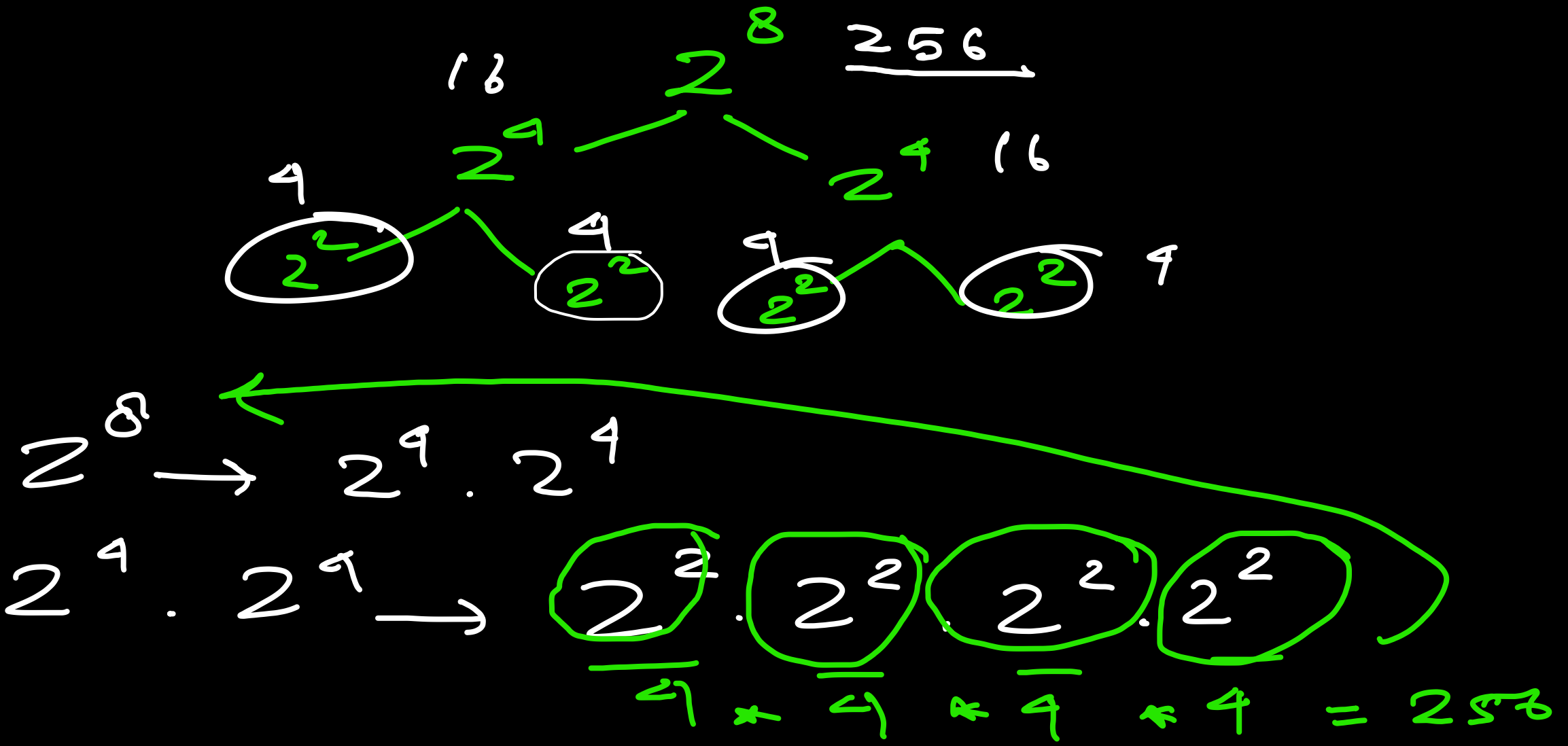
Divide

conquer

Combine







DNC + Memoisation = DP

```
pangkat(a,n):
  if(n == 1) return a
  else return a * pangkat(a,n - 1)
```

$O(N)$

$3^9 = 3 \cdot 3^8$

$3^n = 3 \cdot$

2^{10}

$(2^5)^2$

```
pangkat(a,n):
if(n == 1) return a
else:
    if(n % 2 == 0)
      return pangkat(pangkat(a,n / 2), 2)
    elif(n% 2 == 1)
      return pangkat(pangkat(a,(n - 1) / 2), 2) * a
```

5

$(2^5)^2$

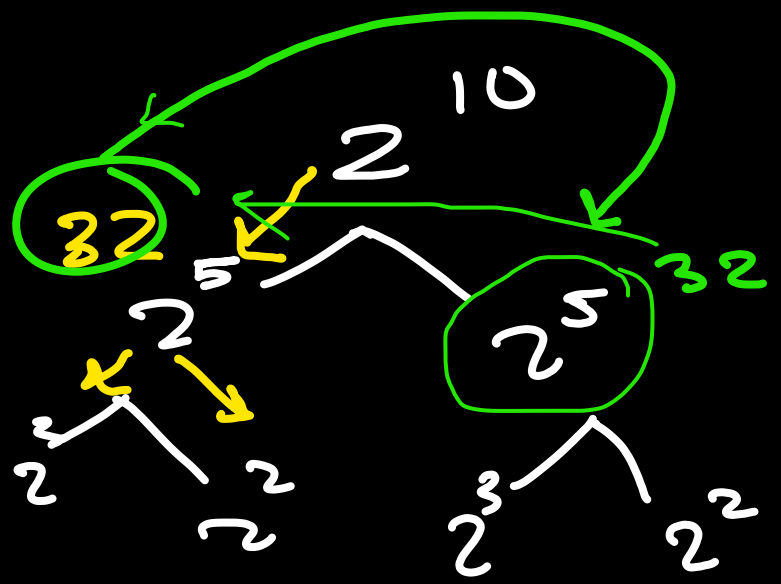
$2^{10} \rightarrow (2^5)^2$

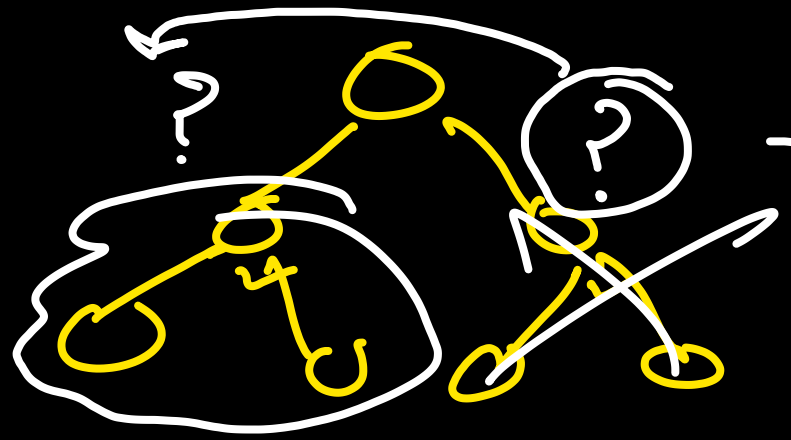
$2^{11} \rightarrow (2^5)^2 \cdot 2^1$

$2^{10} \cdot 2$

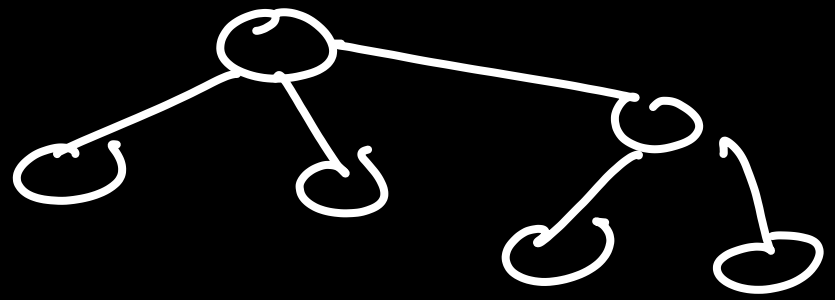
* Memoisation

Memo[5] = 32

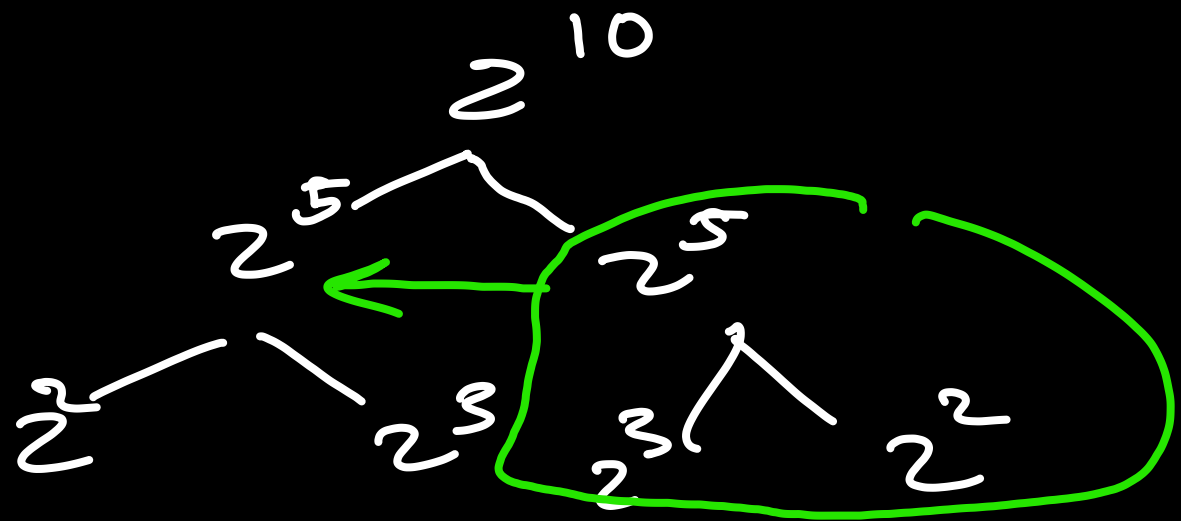


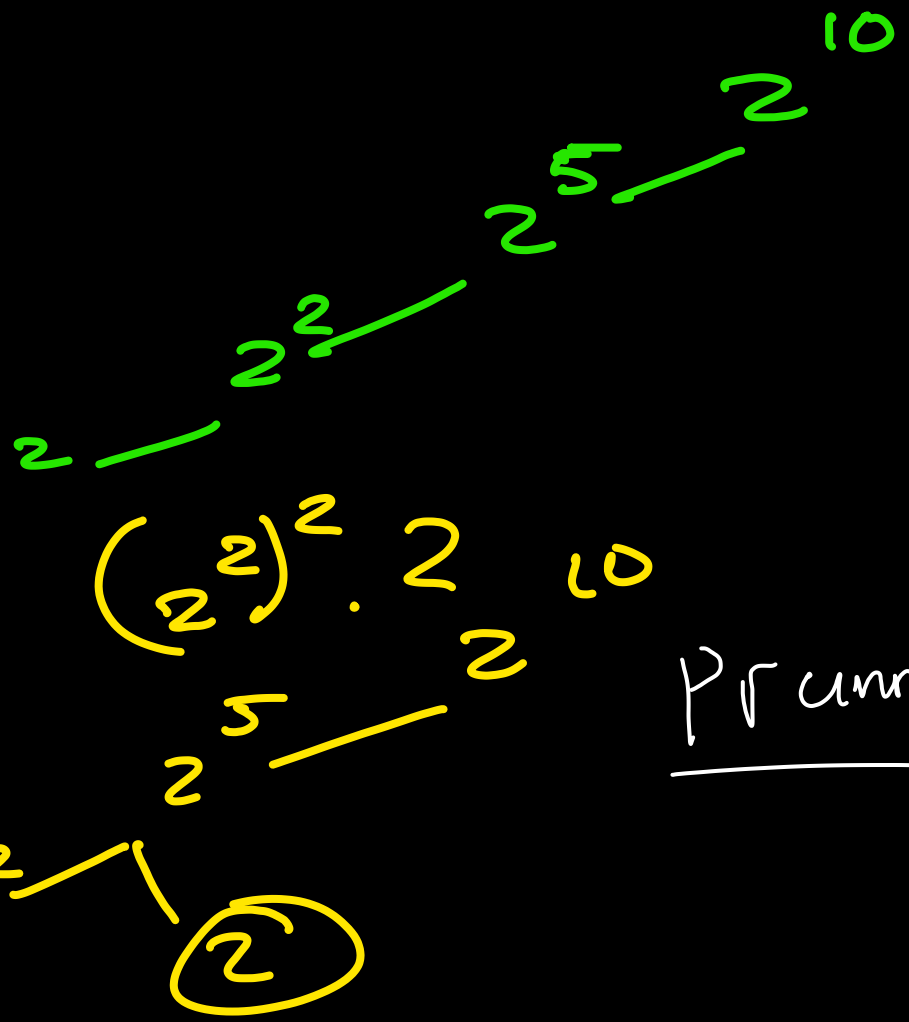


→ memoisation



↑ DnC → Binex




Pruning

1 5 3 4 9 2 ->

min() -> O(n) = 1
sort(arr) -> {1,2,3,4,5,9} , arr[0] = min, arr[n - 1] = max -> O(n * log n)

$1 \rightarrow N \rightarrow \text{sum} = \frac{N * (N+1)}{2}$

Sum seharusnya $= \frac{N * (N+1)}{2}$

Sum array = ...

1 2 0 † 5

O(N)

Sum array = 12

Sum seh = 15

p q r

- if i work its either sunny or partly sunny
- i worked last monday or i worked last Friday
- it wasn't sunny on Tuesday
- it wasn't partly sunny on Friday

∴ Sunny on Friday

Saya sudah bekerja (Past tense) $\equiv p$

$\neg r$

Dom : Friday

$p \wedge \neg r$

$p \rightarrow (q \text{ xor } r)$

p

$\therefore q \text{ xor } r$

$\neg r$

$\therefore q$

Ekivalen / kongmen

$$p \rightarrow q \equiv \neg p \vee q$$

p	q	$p \rightarrow q$	$\neg p \vee q$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

$p \vee q = T$
$T \quad F = T$
$F \quad T = T$
$T \quad T = T$

* Sama kapi tak serupa $\neg p \vee q \equiv T$
$F \vee T$

$Prop_1$ $Prop_2$

Ekivalen saat

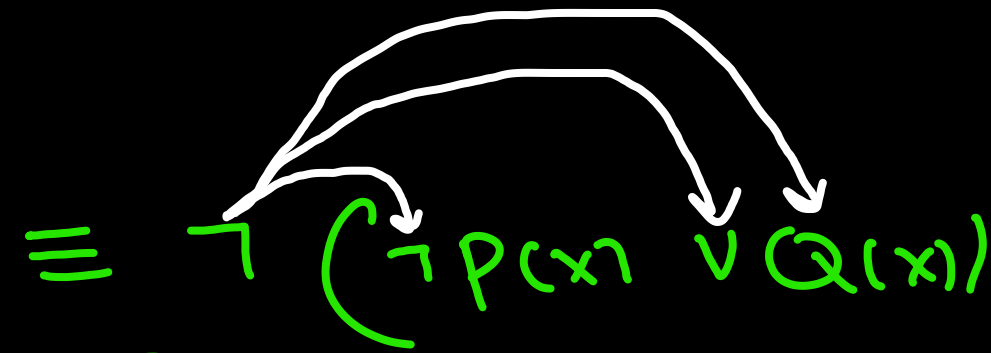
P	q	$P \leftrightarrow q$
T	T	T
F	F	T
T	F	F
F	T	F

$Prop_1 \leftrightarrow Prop_2 \equiv True$

$$\forall_x \left(P(x) \rightarrow Q(x) \right)$$

$$\neg \left(\forall_x \right) \equiv \exists_x$$

$$\neg \left(P(x) \rightarrow Q(x) \right)$$



$$\exists_x \left(P(x) \wedge \neg Q(x) \right)$$

G → O (True)

F → F ≡ T

G jujur, O ~~jujur~~

G, O, L, D

F F T T

Bohong, Jujur

Tidak ada kontradiksi

Semua binatang punya kaki

- x = binatang
- P(x) = x punya kaki

$\forall_x (P(x))$

Tidak ada kelinci yang tau kalkulus

$\equiv$ Semua kelinci (P_1) tidak tau kalkulus (P_2)

x = binatang

Quantifier , Predicate

$\forall_x \quad \exists_x$ $\qquad P(x)$, $Q(x)$, ...

$\Gamma(x)$ = x adalah kelinci

$C(x)$ = x tau kalkulus

$\forall_x (\Gamma(x) \rightarrow \neg C(x))$

x = Binatang

$\neg(\forall_x (\Gamma(x) \rightarrow \neg C(x)))$

X = adalah kunci

C(x) = x tidak lulus kalkulus

✓ $\forall_x \; C \rightarrow \neg C(x)$

$\neg \forall_x \equiv \exists_x$ ✓

$\neg \exists_x \equiv \forall_x$

$P \rightarrow Q \equiv \neg P \vee Q$

$\neg (T(x) \rightarrow \neg C(x))$

$\neg (C \; \neg T(x) \vee \neg C(x)) \equiv T(x) \wedge C(x)$

$\exists x \, C(r(x) \land C(x))$

$P \to q$

$P \leftrightarrow q$

$P \oplus q$

$\land, \vee$

$P \vee q$

$\neg P$

$\therefore q$

silogisme

disjungtif

$\neg(\neg P \vee q)$

Negasi $\equiv P \land \neg q$

Bilimplikasi → xor

Wayan suka makan udang atau ayam -> Wayan suka makan udang atau wayan suka makan ayam
Wayan tidak suka makan udang

:: Wayan Suka makan ayam

$\neg(P \Leftrightarrow q) \equiv (P \land \neg q) \vee (q \land \neg P)$

P = Wayan suka makan udang
Q = Wayan suka makan ayam
Wayan suka makan udang atau ayam -> P V Q

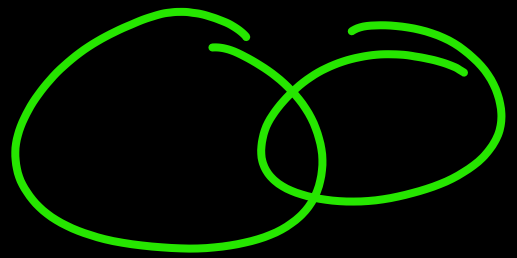
$P \Leftrightarrow q \equiv P == q$ xor

$\neg(P \Leftrightarrow q) \equiv P != q$

Nor xor

Cerah atau sebagian cerah
Tidak Cerah
:: Sebagian Cerah

Logika



p	q	p ∨ q
T	T	T
T	F	T
F	T	T
F	F	F

p ^ q

Hari ini hujan atau cerah

Wayan bekerja di hari Jumat atau Senin