1) Base case - f(0) = ..., f(1) = ...

2) Rekurens - f(n) = ... ?

1) Make a recursive function sum(n) that will do 1 + 2 + 3 + ... + n

1) sum(1) = 1

2) sum(n) = sum(n - 1) + n

sum(1) = 1

sum(2) = 1 + 2 = sum(1) + 2

sum(3) = 1 + 2 + 3 = sum(2) + 3

sum(4) = 1 + 2 + 3 + 4 = sum(3) + 4

...

```
int pentol(int x){
  if(x == 1){
     return 2;
  }else{
     return 1 + pentol(x / 2);
  }
}
```

pentol(16)

P(16) = 1 + P(8) = 2+1+1+1+2

P(8) = 1 + P(4) = 1+1+1+2

P(4) = 1 + P(2) = 1+1+2

P(2) = 1 + P(1) = 1+2

P(1) = 2

$2^{\log 16}$

P(16) = 1 * 4 + 2

P(8) = 1 * 3 + 2

P(4) = 1 * $2^{\log 8}$ + 2

$2\log 4$

$2\log 2$

P(2) = 1 * 1 + 2

P(1) = 2

* logaritma

$$\underbrace{a * a * a * a * \ldots}_{\text{sebanyak } n} * a = a^n$$

$$n / \underbrace{a / a / a / a / a / \ldots}_{\text{sebanyak } x} / a = 1$$

$$\frac{n}{a^x} = 1$$

$$n = a^x$$

$$x = \ldots ? \rightarrow x = a^{\log n}$$

$$2^x = 8 \leftrightarrow x = 2^{\log 8}$$

$$x = 3$$

$$P_{enhol}(x) = 2^{\log x} + 2 \rightarrow \text{explicit}$$

Recursive $\rightarrow$ explicit

Explicit → Recursive

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

→ $Sum(n) = Sum(n-1) + n$

$Sum(1) = 1$

* $Plus(a,b) = a + b$

$Plus(5,1) = 5 + 1$

$Plus(5,2) = 5 + 1 + 1 = P(5,1) + 1$

$Plus(5,3) = 5 + 1 + 1 + 1 = P(5,2) + 1$

$Plus(5,0) = 5$

1) Base - case → $Pls(a,0) = a$

2) Recurrence → $plus(a,b) = Plus(a, b-1) + 1$

* Plus(a,b) = Plus(a,b-1) + 1
  Plus(a,0) = a

Sekarang - sebelumnya

- Buat recursive form / function
1) f(a,b) = a + b → + 1
2) g(a,b) = a - b → - 1
3) h(a,b) = a * b
4) i(a,b) = a / b
5) j(a,b) = power(a,b)

5 * 1 = 5 = 5 + 0
5 * 2 = 10 = 5 + 5
5 * 3 = 15 = 10 + 5
5 * 4 = 15 + 5

h(5,4) = 5 * 4

1) Base case = h(a,1) = a

2) Recurrence Perkalian a * b = Perkalian a * b-1 + a

h(a,b) = h(a,b-1) + a

$i(a,b) = a / b$

$9 / 3 = 9 - 3 - 3 - 3 = 0$

sebanyak 3 kali

$a / b = a - b - b - b - \dots b = 0$

```
if (a==0) return =0
```

$i(9,3) = i(9-3,3) = i(6,3)$

$i(6,3) = i(6-3,3) = i(3,3)$

$i(3,3) = i(3-3,3) = 0$ ✓

$$i(a,b) = 1 + i(a-b, b) \quad \text{if } (a == b) \text{ ret } 1$$

$$O(n * \log n)$$

→ 1 2 3 4 → binser (1) Merge array 1D

→ 5 6 7 8 → binser 10

→ 9 10 11 12

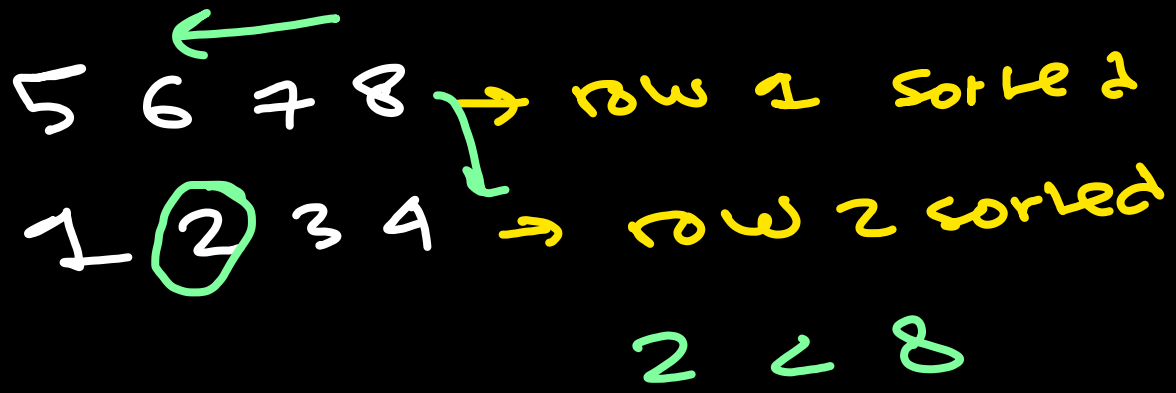
13 14 15 16

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

binser (x) ↑ binary search

$O(M + 2 * \log n)$

↓ Merge M bari's, sort perbari's



M rows, swap row binary → O(m * log n)

- Kasus I : Uang (+), Barang (+)

  Barang : 1 2 3 4

  Uang : 2 4 6 8

  Beli semua barang, bayar pakai uang paling

- Kasus II : Uang (-), Barang (-)

  Barang : -1 -2 -3 -4

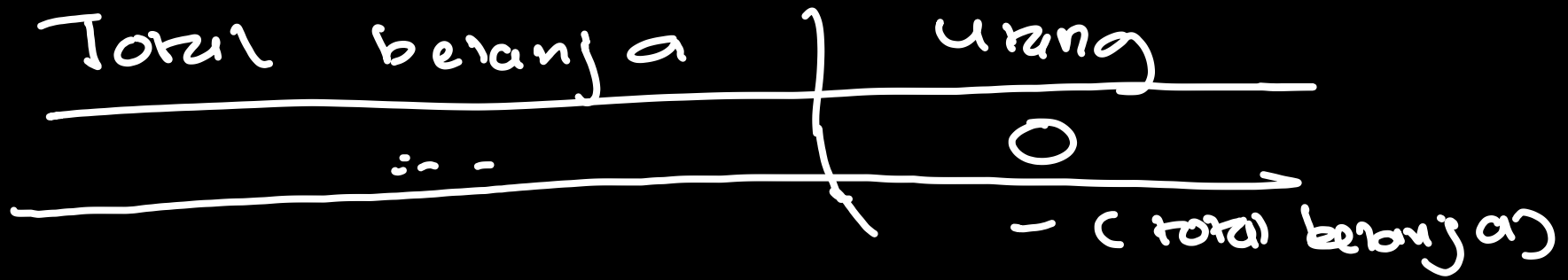
  Uang : -1 -2 -3 -4

  } Barang max, Bayar pakai semua uang

- Kasus III : Uang (+), Barang (-)

  Utang = 0 , belanja = uang - barang

  = 3



* Kasus III : uang (-), barang (+)

beli semua barang, bayar pakai' semua uang

← kasus II : → uang (variatif), barang (variatif)

Pilih barang (+), uang (-)

→ uang (-), barang (variatif)

→ uang (variatif), barang (+)

kasus VI : → uang (+), barang (variatif)

bayar pakai uang min, barang (+)

→ uang (variatif), barang (-)

barang ambil min, uang ambil (-)