

- 1) Base case - $f(0) = \dots, f(1) = \dots$
- 2) Rekurens - $f(n) = \dots ?$

1) Make a recursive function sum(n) that will do $1 + 2 + 3 + \dots + n$

1) $\text{sum}(1) = 1$

2) $\text{sum}(n) = \text{sum}(n-1) + n$

$$\text{sum}(1) = 1$$

$$\text{sum}(2) = \underline{1} + 2 = \text{sum}(1) + 2$$

$$\text{sum}(3) = 1 + 2 + 3 = \text{sum}(2) + 3$$

$$\text{sum}(4) = 1 + 2 + 3 + 4 = \text{sum}(3) + 4$$

...

```

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

```

pentol(16)

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

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

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

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

$$P(1) = 2$$

$2^{\log 16}$

$$P(16) = 1 + \overbrace{4}^{2^{\log 4}} + 2$$

$$P(8) = 1 + \overbrace{3}^{2^{\log 8}} + 2$$

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

$$P(2) = 1 + \overbrace{1}^{2^{\log 2}} + 2$$

$$P(1) = 2$$

* Logaritma



$$a * a * a * a * \dots + a = a^n$$

sebanyak n

$$\underbrace{a/a/a/a/a/\dots/a}_n = 1$$

sebanyak

x

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

$$n = a^x$$

$$x = \dots ? \rightarrow x = a \log n$$

$$2^x = 8 \Leftrightarrow x = 2 \log 8$$

$$x = 3$$

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

Recursive \rightarrow explicit

Explicit \rightarrow Recursive

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

* $\text{Plus}(a, b) = a + b$

$$\text{Plus}(5, 0) = 5$$

$$\text{Plus}(5, 1) = 5 + 1$$

$$\text{Plus}(5, 2) = 5 + 1 + 1 = \text{Plus}(5, 1) + 1$$

$$\text{Plus}(5, 3) = 5 + 1 + 1 + 1 = \text{Plus}(5, 2) + 1$$

1) Base-case $\rightarrow \text{Plus}(a, 0) = a$

2) Recurrence $\rightarrow \text{Plus}(a, b) = \text{Plus}(a, b-1) + 1$

$$+ \text{Plus}(a, b) = \text{Plus}(a, b-1) + 1 \}$$

$$\text{Plus}(a, 0) = a \quad \text{sekarang - sebelumnya}$$

- Buat recursive form / function

$$1) f(a, b) = a + b \rightarrow +1$$

$$2) g(a, b) = a - b \rightarrow -1$$

$$3) h(a, b) = a * b$$

$$4) i(a, b) = a / b$$

$$5) j(a, b) = \text{power}(a, b)$$

$$\begin{aligned}
 & 5 + 1 = 5 = 5 + 0 \\
 & 5 + 2 = 10 = 5 + 5 \\
 & 5 + 3 = 15 = 10 + 5 \\
 & \underline{5 + 4} = 15 + 5
 \end{aligned}$$

$$h(5, 4) = 5 + 4$$

$$1) \text{Base case} = h(a, 1) = a$$

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

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

$$i(a,b) = a \text{ / } b$$

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

sebanyak 3 kali

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

if ($a == 0$) return 0

$$i(9, \underline{3}) = i(9 - 3, 3) = i(6, 3)$$

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

$$i(3, 3) = i(3 - 3, 3) = 0 \quad \checkmark$$

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

ref 1

$O(n * \log n)$

$\rightarrow 1 \ 2 \ 3 \ \boxed{4} \rightarrow \text{binser}$ (1) Merge array 1D
 $\rightarrow 5 \ 6 \ 7 \ \boxed{8} \rightarrow \text{binser}$ 1D
 $\rightarrow 9 \ 10 \ 11 \ 12$
 $13 \ 14 \ 15 \ 16$

$1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ \dots \ 16$
 binser (\times) \uparrow binary search
 $O(M+2 * \log n)$

\downarrow
 merge n bin's, sort
 per bin's

5 6 7 8 ← row 1 sorted
1 2 3 9 → row 2 sorted
2 < 8

M rows, search row binary $\rightarrow O(n + \log n)$

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

Barang :

1	2	3	4
---	---	---	---

uang :

2	4	6	8
---	---	---	---

Beli semua barang, bayar pakai uang pasiung

- * Kasus II : uang (-), barang (-)

Barang :

-1	-2	-3	-4
----	----	----	----

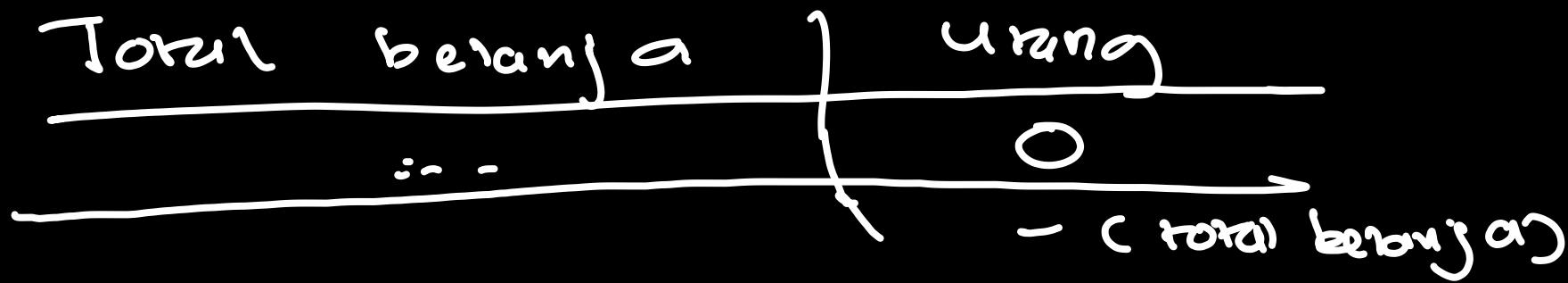
uang :

-1	-2	-3	-4
----	----	----	----

Barang mat,
Bayar pakai semua
uang

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

uang = 0, belanja = uang-barang
= 3



- * Kasus I : uang (-), barang (+)
beli semua barang bayar pakai semua uang
- * Kasus II : → uang (varianif), barang (varianif)
pilih barang (+), uang (-) ←
→ uang (-), barang (varianif)
→ uang (varianif), barang (+)

vars ∇ : $\rightarrow \frac{\text{uang} (+), \text{barang} (\text{variatif})}{\text{bayar barang uang min.}, \text{barang} (+)}$
 $\rightarrow \text{uang} (\text{variatif}), \frac{\text{barang} (-)}{\text{barang ambil min}, \frac{\text{uang ambil} (-)}{}}$

```

void h(int N){
    if(N == 0) {
        cout<<"";
    }else{
        h(N - 1);
        cout<<N;
        h(N - 1);
    }
}

```

