No.

Date . .

Dengan ini saya menyatakan bahwa PR ini adalah hasil pekerjaan saya sendiri

ALDENLUTHFI

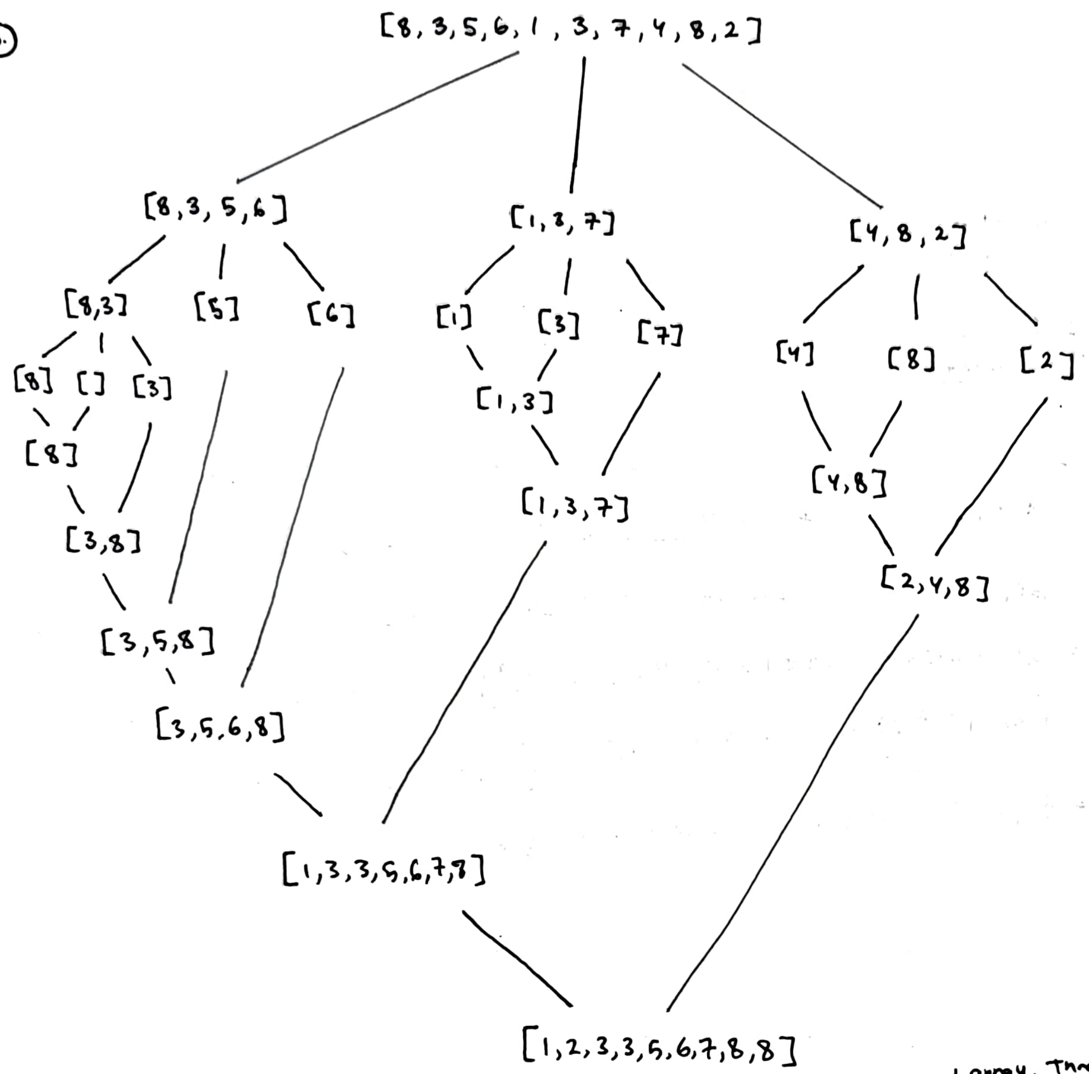
ALDEN LUTHFI - 2206028932

1.) a.

```
mergeSort2(A, p, r):
    if p < r:
        n = r - p
        q1 = p + ⌊n/3⌋
        q2 = p + ⌊2n/3⌋
        mergeSort2(A, p, q1)
        mergeSort2(A, q1+1, q2)
        mergeSort2(A, q2+1, r)
        merge(A, p, q1, q2)
        merge(A, p, q2, r)
```

→ fungsi merge() tidak diubah

b.




c.) Algoritma sorting disebut stable jika urutan elemen yang bernilai sama sesuai dengan input array. Three way merge sort ini stable karena mendahulukan array yang lebih kiri jika ada yang bernilai sama.

d.) Algoritma sorting disebut in place jika tidak mengalokasikan memori untuk array baru. Three way merge sort ini tidak in place karena butuh array baru di proses merge()

e.) $$T(n) \begin{cases} 1 & 0 \leq n \leq 1 \\ 3T\left(\frac{n}{3}\right) + n & n > 1 \end{cases}$$

f.) dengan Master's Theorem

↳ $a = 3$ ↳ $k = 1$

↳ $b = 3$ ↳ $p = 0$

Case 2

$p > -1 \rightarrow T(n) = \Theta(n \log n)$

No.

Date

Revisi 1 f :

$$T(n) \begin{cases} 1 \\ 3T(\frac{n}{3}) + n \end{cases}$$

PROOF BY INDUCTION

$T(n) = O(n^{3}\log n)$

$P(n) : \exists c \, (T(n) \leq cn \log_3 n)$

Base Induction :

↳ $P(3) : \exists c \, (T(3) \leq c3 . \log_3 3) \rightarrow \exists c \, ( 6 \leq c3$

BENAR ketika $c \geq 2$

Inductive State .

↳ asumsi $P(\frac{n}{3}) : \exists c \, (T(\frac{n}{3}) \leq c . \frac{n}{3} \log_3 \frac{n}{3})$ BENAR

↳ $P(n)$ BENAR ~~maka~~ karena

$$\begin{aligned} T(n) &= 3T(\tfrac{n}{3}) + n \\ &\leq 3 . c . \tfrac{n}{3} \log_3 \tfrac{n}{3} + n \\ &= cn(\log_3 n - \cancel{\log}\, 1) + n \\ &= cn\log_3 n - cn + n \\ &\leq n\log_3 n \quad \text{karena} \quad -cn + n \leq 0 \text{ saat } c \geq 2 \end{aligned}$$

maka terbukti $T(n) = O(n\log_3 n)$

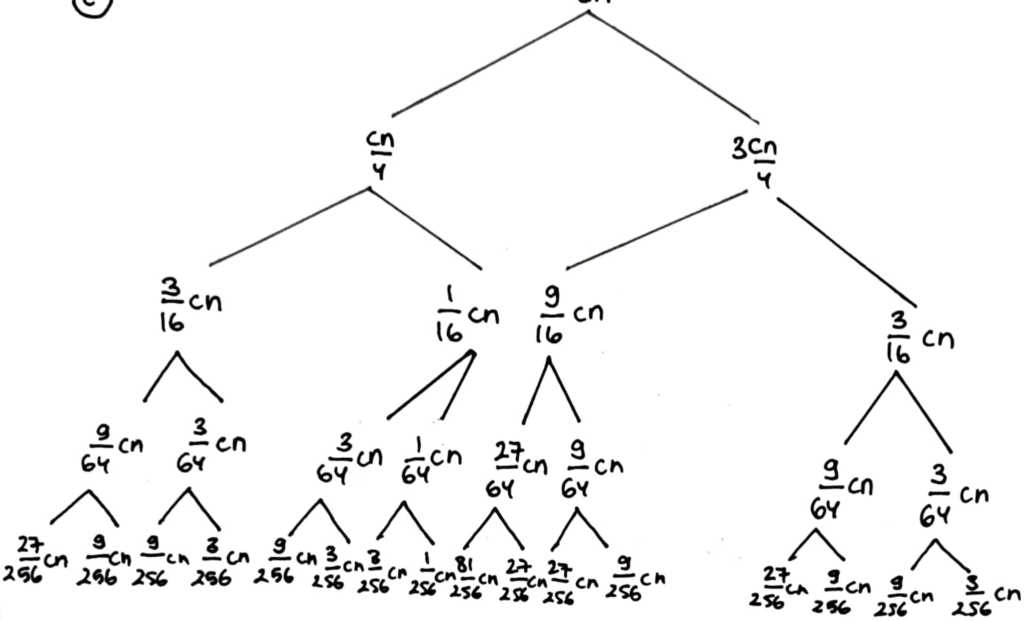
ALDEN LUTHFI - 2206028932

2. a. $T(n) = T(\frac{1}{4}n) + T(\frac{3}{4}n) + \theta(n)$

b. tinggi tree : $^{\frac{4}{3}}\log n$

banyak level : $^{\frac{4}{3}}\log n + 1$

c.




| dominant term | $\Sigma$ cost |
|---|---|
| $T(n)$ | cn |
| $T(\frac{3n}{4})$ | cn |
| $T(\frac{9}{16}n)$ | cn |
| $T(\frac{27}{64}n)$ | cn |
| $T(\frac{81}{256}n)$ | cn |

d. tree tidak akan penuh untuk level $\geq$ $^4\log n$ karena term $T(\frac{n}{4})$ ... sudah habis

e. karena ada $^{\frac{4}{3}}\log n + 1$ level dan cn cost per level

$T(n) = \theta(n \log n)$

3. a. $T(n) = 4T(\frac{n}{2}) + n^2 \lg n$

$T(n) = O(n^2 \lg^2 n)$ } asumsi $T(1) = 1$

PROOF BY INDUCTION

$P(n) : \exists c (T(n) \leq c . n^2 \lg^2 n)$

Base Case :

↳ $P(2)$ : $\exists c (T(2) \leq c . 4)$

→ $8 \leq 4c$ untuk $c \geq 2$

Induction Case :

↳ $P(\frac{n}{2}) : \exists c (T(\frac{n}{2}) \leq c . \frac{n^2}{4} \lg^2 \frac{n}{2})$ diasumsikan BENAR

↳ $P(n)$ BENAR karena

$$
\begin{aligned}
T(n) &= 4T(\tfrac{n}{2}) + n^2 \lg n \\
&\leq 4(\tfrac{n^2}{4} \lg^2 \tfrac{n}{2}) + n^2 \lg n \\
&= n^2 \lg^2 \tfrac{n}{2} + n^2 \lg n \\
&= n^2 \lg^2 n + (-2n^2 \lg n + n^2 + n^2 \lg n) \\
&= n^2 \lg^2 n + (n^2 - n^2 \lg n) \\
&\leq n^2 \lg^2 n \quad \text{karena } n^2 - n^2 \lg n \leq 0 \text{ untuk } n \geq 2
\end{aligned}
$$

ALDEN LUTHFI - 2206028932

(b.) $T(n) = 2T(\frac{n}{2}) + \frac{n}{\lg n}$

↳ $a = 2$

↳ $b = 2$

↳ $E = \lg 2 = 1$

↳ $f(n) = n \lg^{-1} n = \Theta(n^E (\log_2 n)^{-1})$

Case 2

↳ maka $T(n) = \Theta(n \lg(\lg n))$

(c.) $T(n) = 4T(\frac{n}{3}) + n$

↳ $a = 4$

↳ $b = 3$

↳ $f(n) = n = \Theta(n)$

$a > b^k \rightarrow 4 > 3^1$

Case 1

$T(n) = \Theta(n^{^3\log 4})$

(d.) $T(n) = 2T(\frac{n}{4}) + \sqrt{n}$

↳ $a = 2$

↳ $b = 4$

↳ $f(n) = n^{\frac{1}{2}} = \Theta(n^{\frac{1}{2}})$

$a = b^k \rightarrow 2 = 4^{\frac{1}{2}}$ , $p > -1$

Case 2

maka $T(n) = \Theta(\sqrt{n} \lg n)$

Master's Theorem Extended

$T(n) = aT(\frac{n}{b}) + \Theta(n^k \log^p n)$

1) $a > b^k \rightarrow T(n) = \Theta(n^{^b\log a})$

2) $a = b^k$

↳ $p > -1 \rightarrow T(n) = \Theta(n^{^b\log a} \log^{p+1} n)$

↳ $p = -1 \rightarrow T(n) = \Theta(n^{^b\log a} \log(\log n))$

↳ $p < -1 \rightarrow T(n) = \Theta(n^{^b\log a})$

3) $a < b^k$

↳ $p \geq 0 \rightarrow T(n) = \Theta(n^k \log^p n)$

↳ $p < 0 \rightarrow T(n) = \Theta(n^k)$

(4.) (a.) nf = 5

ng = 5

nres = 9

res = [0, 0, 0, 0, 0, 0, 0, 0, 0]

fp = [0, 2.5, 5.0, 7.5, 10.0, 0, 0, 0, 0]

gp = [2, 4, 6, 8, 10, 0, 0, 0, 0]

→ n = 0

↳ m = 0 : res[0] += 2 * 0

res = [0, 0, 0, 0, 0, 0, 0, 0, 0]

→ n = 1

↳ m = 0 : res[1] += 4 * 0

↳ m = 1 : res[1] += 5.0

res = [0, 5.0, 0, 0, 0, 0, 0, 0, 0]

→ n = 2

↳ m = 0 : res[2] += 0

↳ m = 1 : res[2] += 10

↳ m = 2 : res[2] += 10

res = [0, 5, 10, 0, 0, 0, 0, 0, 0]

→ n = 3

↳ m = 0 : res[3] += 0

↳ m = 1 : res[3] += ~~20~~ 15

↳ m = 2 : res[3] += 20

↳ m = 3 : res[3] += 15

res = [0, 5, 10, 50, 0, 0, 0, 0, 0]

→ n = 4

↳ m = 0 : res[4] += 0

↳ m = 1 : res[4] += 20

↳ m = 2 : res[4] += 30

↳ m = 3 : res[4] += 30

↳ m = 4 : res[4] += 20

res = [0, 5, 10, 50, 100, 0, 0, 0, 0]

→ n = 5

↳ m = 0 : res[5] += 0

↳ m = 1 : res[5] += 25

↳ m = 2 : res[5] += 40

↳ m = 3 : res[5] += 45

↳ m = 4 : res[5] += 40

↳ m = 5 : res[5] += 0

res = [0, 5, 10, 50, 100, 150, 0, 0, 0]

→ n = 6

↳ m = 0 : res[6] += 0

↳ m = 1 : res[6] += 0

↳ m = 2 : res[6] += 50

↳ m = 3 : res[6] += 60

↳ m = 4 : res[6] += 60

↳ m = 5 : res[6] += 0

↳ m = 6 : res[6] += 0

res = [0, 5, 10, 50, 100, 150, 170, 0, 0]

→ n = 7

↳ m = 0 : res[7] += 0

↳ m = 1 : res[7] += 0

↳ m = 2 : res[7] += 0

↳ m = 3 : res[7] += 75

↳ m = 4 : res[7] += 80

↳ m = 5 : res[7] += 0

↳ m = 6 : res[7] += 0

↳ m = 7 : res[7] += 0

res = [0, 5, 10, 50, 100, 150, 170, 155, 0]

→ n = 8

↳ m = 0 : res[8] += 0

↳ m = 1 : res[8] += 0

↳ m = 2 : res[8] += 0

↳ m = 3 : res[8] += 0

↳ m = 4 : res[8] += 100

↳ m = 5 : res[8] += 0

↳ m = 6 : res[8] += 0

↳ m = 7 : res[8] += 0

↳ m = 8 : res[8] += 0

res = [0, 5, 10, 50, 100, 150, 170, 155, 100]

→ return res

ALDEN LUTHFI - 2206028932

(b.)

```
   discrete_convolution(f, g):
1      nf = length(f)
2      ng = length(g)
3      nres = nf + ng - 1
4      let res[0.. nres-1] be new arrays
5      fp = f + [0] * (nres - nf)
6      gp = g + [0] * (nres - ng)
7      for n in range(nres):
8          for m in range(n+1):
9              res[n] += fp[m] * gp[n-m]
10     return result
```

ASUMSI :

$C_1$ = assignment

$C_2$ = arith. addition / substraction

$C_3$ = arith. multiplication

$C_4$ = array concatenation

Array Initialization has no cost

| Line | Cost |
| --- | --- |
| 1 | $C_1$ |
| 2 | $C_1$ |
| 3 | $C_1 + 2C_2$ |
| 4 | $C_1$ |
| 5 | $C_1 + C_2 + C_4(nres - nf + 1)$ |
| 6 | $C_1 + C_2 + C_4(nres - ng + 1)$ |
| 7 | $C_1\, nres$ |
| 8 | $C_1\left(\frac{nres(nres+1)}{2}\right)$ |
| 9 | $(C_1 + C_2 + C_3)\left(\frac{nres(nres+1)}{2}\right)$ |
| 10 | 0 |

$+$

$$\text{total cost} = C_1 + C_1 + C_1 + 2C_2 + C_1 + C_1 + C_2 + C_4(nres - nf + 1) + C_1 + C_2 + C_4(nres - ng + 1) + C_1\, nres + C_1\frac{nres(nres+1)}{2} + C_1\frac{nres(nres+1)}{2} + C_2\frac{nres(nres+1)}{2} + C_3\frac{nres(nres+1)}{2}$$

$$= 6C_1 + C_1 nres + C_1 nres(nres+1) + 4C_2 + C_2\frac{nres(nres+1)}{2} + C_3\frac{nres(nres+1)}{2} + C_4(nres + nres - (nf+ng) + 2)$$

$$= C_1(6 + nres + nres^2 + nres) + C_2\left(4 + \frac{nres}{2} + \frac{nres^2}{2}\right) + C_3\left(\frac{nres^2}{2} + \frac{nres}{2}\right) + C_4(nres + 1)$$

$$= nres^2\left(C_1 + \frac{C_2}{2} + \frac{C_3}{2}\right) + nres\left(2C_1 + \frac{C_2}{2} + \frac{C_3}{3} + C_4\right) + (6C_1 + 4C_2 + C_4)$$

$= \Theta(nres^2)$ dengan $nres$ = length(f) + lenght(g) - 1